AFRL-IF-RS-TR-2005-190
Final Technical Report
May 2005

REAL-TIME CONFIGURATION OF NETWORKED EMBEDDED SYSTEMS

University of Notre Dame

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. L546

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-190 has been reviewed and is approved for publication

APPROVED: /s/

RAYMOND A. LIUZZI
Project Engineer

FOR THE DIRECTOR: /s/

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** MAY 2005	**3. REPORT TYPE AND DATES COVERED** Final May 01 – Dec 04

4. TITLE AND SUBTITLE

REAL-TIME CONFIGURATION OF NETWORKED EMBEDDED SYSTEMS

5. FUNDING NUMBERS

C - F30602-01-2-0526
PE - 62301E
PR - NEST
TA - 00
WU - 01

6. AUTHOR(S)

Raymond A. Liuzzi, Panos Antsaklis et al.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Notre Dame
511 Main Building
Notre Dame Indiana 46556

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFED
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2005-190

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Raymond A. Liuzzi/IFED/(315) 330-3577/ Raymond.Liuzzi@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

This project focused on the development of synchronization algorithms and software for a network of MICA (or Berkeley) motes. Automata and Petri net models of operational aspects were derived and verification techniques were used to guarantee correctness. A major part of this project was dedicated tot the Red Force Tagging or Tag Mote) project. This report describes detailed descriptions of the developed TagMote hardware, software and algorithms. This report also describes fundamental contributions to the communications in wireless sensor networks. The networking results address issues brought forth because of the increasing number of sensor nodes in applications of interest. Also in this project, new results are described on overload management of distributed control to address problems arising in the fairing application of Boeing. In summary, the first part of this report describes the Red Force Tagging project, while the second part describes the Networking research results. The appendices contain further information and copies of publications. Copies of PI presentations are also included.

14. SUBJECT TERMS

Networks, Verification, Software, Wireless Sensor Networks

15. NUMBER OF PAGES

424

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

OVERVIEW

This project initially focused on the development of synchronization algorithms and software for a network of embedded MICA (or Berkeley) motes. Automata and Petri net models of operational aspects were derived and verification techniques were used to guarantee correctness. These results were reported in the PI presentations, copies of which accompany this report (Appendix D). A major part of this project was dedicated to the Red Force Tagging project. Since the Red Force Tagging (or TagMote) project incorporated the earlier network middleware developed in this project, in this final report we have primarily focused on describing in full detail the TagMote project.

Specifically, in the first part of this report, there is a detailed description of the developed TagMote hardware, software and algorithms. In the second part, fundamental contributions to the communications in wireless sensor networks are described with most of the technical details included in the publications, copies of which are included in the Appendices. The networking results were developed for the most part during the second half of this project to address issues brought forth, because of the increasing number of sensor nodes in the applications of interest. Note that in this project additional new results on overload management for distributed control were developed to address problems arising in the fairing application of Boeing (see Appendix C).

In summary, the first part of this report describes the Red Force Tagging project, while the second part describes the Networking research results. The Appendices contain further information and copies of publications. Electronic copies of the PI presentations also accompany this report.

DARPA IXO/NEST Award No. F30602-01-2-0526

Final Report

Part I

Red Force Tagging

Section 1. Executive Summary

In this section the goals and accomplishments of the Red Force Tagging project (TagMote project) are briefly described. High level descriptions of the TagMote operation, the flow of information, the hardware, middleware and algorithms are given. Appendix D contains copies of the PI meeting presentations where the TagMote hardware, software and operation are described in detail.

The goal of this TagMote project was the reconstruction of the space/time path a subject has taken from an initial to a final location. The reconstruction is based on Dead Reckoning (DR) and it works in areas where GPS is not available, for example in urban canyons and inside buildings. Such information is also useful to civilians, as it can be used for personal navigation by campers and hikers, firemen and police. It addition, such capability can be used for mapping an area, specifically to create maps of areas where GPS is not available, by just having someone walk through these areas. The reconstruction algorithms only use data collected by a special MEMS device the subject has been carrying. When the subject arrives in the proximity of a deployed sensor network, the data collected on the portable device are extracted automatically via wireless communication and deposited in the nodes of the sensor network. These data that are distributed within the network are then sent to a collection point within the network for further processing and display.

The specific objective of the TagMote project was to demonstrate the feasibility of building such mapping device (hardware, software) that cooperates with the NEST sensor network and utilizes middleware to accomplish the stated goals. This objective was accomplished.

Our approach to pedestrian navigation relies on a specially built MEMS based device (called the TagMote) that includes sensors, memory, a processor and RF communication capabilities. This device uses accelerometers and a digital magnetic compass to collect data regarding distance traveled and direction, and it uses memory to store data points. The device incorporates Berkeley motes (Mica2) to provide processing power and communications. Special software implements algorithms that provide compensation for adverse sensor orientation and calibration, and for collection with possible coding of data and storage on a nonvolatile memory. The data collection and part of the processing takes place while the device carried by the subject is out of range of the network. When the subject enters again into radio range of the network, a communication link is established automatically and the data are transmitted initially to one or more of the network motes and then via multi-hop transmissions to a designated special "relay" or "base node" mote for further processing. Algorithms detect occurrence of steps and calculate distance traveled, and use direction of movement and distance traveled to generate trajectory points, which are then appropriately displayed.

There were several successful demonstrations of the project each involving the following steps:

1. Setup: The sensor network grid was physically deployed, initialized and synchronized, so it could receive data from a mobile mote (TagMote) that joins the network, and also could route data, that have been downloaded at any mote, to the designated special relay mote for further processing and display.
2. Initialization: During initialization, the special device (TagMote) acquires its position and time to be used as reference. For the field experiment this was hand programmed into the device.
3. Operation: A subject carrying the special TagMote walked out of range of the network for some time interval, walked in a normal fashion tracing some path and returned within range of the network at some later time.
4. Data transmission to the network: The data from the TagMote were downloaded wirelessly to network motes.
5. Data routed to the relay mote within the network.
6. Data processing, display and evaluation: There is initial data processing on the TagMote before they are being transmitted to the network. In addition, data transmitted to the network and sent up to the relay mote are processed and displayed to show the trajectory traced by the subject.

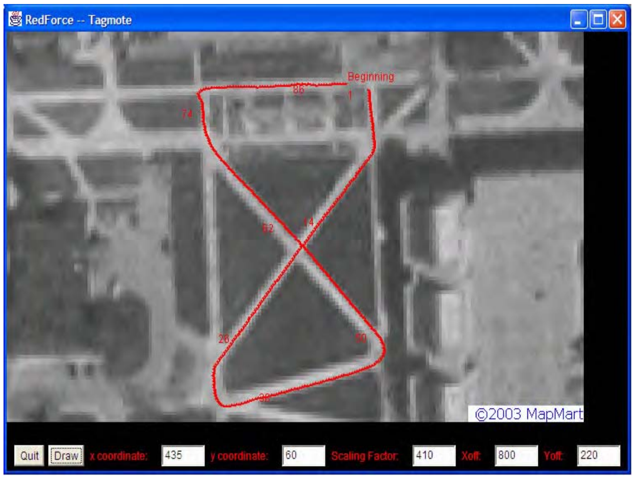
Hardware: We used a specially designed and built MEMS device (TagMote) to mount sensors so to collect data, process and store these data and then transmit to the network. We have designed and built boards that contain a 4MB flash memory, the sensors and additional processors. This TagMote board attaches to a mote as a daughter board, similarly to Mica's sensor boards. We used an integrated 3D-accelerometer and 3D-magnetic compass device that incorporates amplification, signal conditioning, temperature compensation etc made by Leica Geosystems. In this prototype version, the TagMote and Mica boards were packaged together and were attached to the belt of a subject; the battery that supports the TagMote and Mica are packaged separately and are placed in a pocket of the subject.

We are reading six sets of sensor data (3 sets from the accelerometer and 3 sets from the magnetic compass) using a Mica processor and we are storing them in the 4MB flash memory of the TagMote board. We read the data from the flash memory and we send them to the Mica communication stack for transmission to the network in response to an acknowledgement of the TagMote's Ping signal by the network. Just before sending the data to the network we are performing some preprocessing to identify the points where min and max occur in the vertical acceleration and we then only transmit data corresponding to these trajectory points. Later versions, described in detail in the hardware section, involve two processors and more sophisticated processing.

Software: Special software was developed to implement algorithms for processing the sensor data on the mote and before displaying. Algorithms must detect occurrence of steps and calculate distance based on some model for the average stride and compensate for differences.
In general, the larger the amount of processing performed onboard the TagMote, the smaller the amount of time required for downloading the data necessary to display certain distance traveled by the subject.

The amount of processing performed onboard the TagMote is of course determined by the processing power available. In our initial demos (August 2003), we were limited by the single Mica processor. In a later TagMote version, additional processing power was provided by an additional Mica processor and the processing capabilities were significantly enhanced.

After physical deployment of the sensor grid, the network must autonomously reconfigure itself to form a multi-hop communication network in which all nodes have a "global" sync time. The following middleware services are used in the network: Neighborhood initialization so to build tables of neighborhood ID's; Network clock synchronization; Localization (needed to determine TagMote's initial position)-for the demos the initial position and time was introduced manually into the TagMote. The TagMote uses a Ping Service, which is used by the TagMote to find the network; a Data Dump Service is used by TagMote to download (dump) data to the network. The services need to inform the TagMote if a transmitted data packet has been handed off to the network successfully and this is done via an acknowledgement. At the completion of the download all the TagMote data are stored in one or more than one motes of the network. Having more than one of the network motes being involved with the TagMote data download is desirable, since a mote may only have limited memory available and also the subject may be moving from being close to one mote to being closer to a different mote. Such scheme provides enhanced flexibility and robustness. The data stored in the network motes are then transmitted to a relay or base mote typically through multi-hop transmissions. The middleware services need to provide reliable, high throughput channels for streaming TagMote data to the destination. These were provided by A Broker Service that builds a spanning tree back to the relay, and a Publication Service that routes dumped TagMote data to the relay mote.



x coordinate: 435 y coordinate: 60 Scaling Factor: 410 Xoff: 800 Yoff: 220

Figure 1.1 The display of a truth trajectory against a satellite photo

Section 2. Dead Reckoning Algorithm: Errors and Solutions

This section describes the algorithms used to process accelerometer and compass data and estimate position using Dead Reckoning. The algorithms estimate the length of step, direction and step count.

A Dead Reckoning (DR) algorithm estimates the total distance from a known starting point by estimating the size of each step and detecting the heading.

$$P(i+1) = P(i) + d(i) \cdot \vec{a}(i)$$

where $P(i)$: Position after i steps; $d(i)$: Step length of i^{th} step; $\vec{a}$ (i): Magnetic azimuth of i^{th} step.

2.1. Estimation of the Step Length

The 3-axis accelerometer on the waist belt is used to detect acceleration of body movement. From the filtered signal, we can approximate the step length by [1] (reference numbers refer to the list at the end of the present section):

$$d = K \cdot \sqrt[4]{A_{\max} - A_{\min}}$$

where: $A_{\min}$ is the minimum acceleration measured in the Z axis in a single stride; $A_{\max}$ is the maximum acceleration measured in the Z axis in a single stride; K is a constant (that also performs unit conversion i.e. feet or meters traveled).

This technique has been shown to measure distance walked accurately to within ±3% for the same subject and ±8% across a number of subjects with different leg lengths (See Tables 2.1, 2.2 and 2.3 for experiments conducted at Notre Dame). Close coupling of the accelerometer to the body is important to maintain accuracy.

Errors and Solutions:

- We look at the leg as being a lever of fixed length while the foot is on the ground. The experimental results demonstrate that a constant can be used for *K* without a large accuracy penalty. GPS or distributed sensor networks with the localization capability can be utilized to obtain more accurate step length estimation.
- The same *K* can be used for normal, slow and fast walking on different surfaces. A shorter distance will be obtained for fast walking and a larger variance for soft surface such as grassland. The errors are introduced by the violation of our assumption that the leg is fixed length. An adaptive algorithm that "learns" the users stride characteristics could improve the accuracy.

Table 2.1: The constant factor K for three different subjects (75m straight walking on cement sidewalk)

Subject	A	B	C
Leg Length	0.94	1.06	1.08
K (Nominal Value)	0.50	0.55	0.57

Table 2.2. Using the same K to estimate the distance with different walking patterns (75m straight walking on cement sidewalk)

Subject	Normal	Slow	Fast
A (*K* = 0.5)	75.3	74.2	73.5
B (*K* = 0.55)	75.5	74.6	73.3
C (*K* = 0.57)	74.9	76.0	72.8

Table 2.3. The estimated walking distances and errors (East/West Component) when the same subject walked normally towards South/North for 100 meters at a Olympic stadium (with K =0.5)

Case Number	Estimated Distance (m)	Error 1 (m) (no calibration)	Error 2 (m) (with tilt calibration)	Error 3 (m) (with total calibration)
1	102.8	20.2	5.2	2.5
2	101.6	20.5	5.5	2.9
3	102.4	20.8	5.3	2.7
4	101.6	8.4	1.4	1.2
5	102.4	9.3	1.8	0.8
6	103.0	11.1	2.0	0.7
7	102.5	9.9	5.2	2.3
8	102.2	9.3	1.7	0.9
9	102.3	10.4	1.0	1.6
10	102.1	9.0	0.7	1.9
11	100.1	2.7	12.2	0.5
12	103.0	3.2	9.7	0.9
13	102.3	3.3	9.5	0.9
14	100.7	4.8	10.3	2.5
15	102.6	5.3	11.2	3.0
16	103.0	5.6	11.8	3.3
17	101.6	4.4	10.5	2.0
18	102.4	5.6	11.8	3.3
19	102.3	4.8	11.1	2.4
20	102.2	5.3	10.8	3.0
Mean	102.2	6.8°	2.1°	1.1°
STD	0.7	2.1°	1.0°	0.5°

2.2. Compass Navigation

The magnetic azimuth is the horizontal component of the Earth magnetic field. Its determination requires implicitly the knowledge of the horizontal or vertical plane. This is commonly done by sensing the gravity vector at rest. To compute then the azimuth of walk, one has to constantly compute the attitude of the sensor in order to correct the measured magnetic values. Using the 3D rotation matrix with the Yaw(ψ)-Pitch(φ)-Roll(θ) sequence, the horizontal components H_y and H_x are

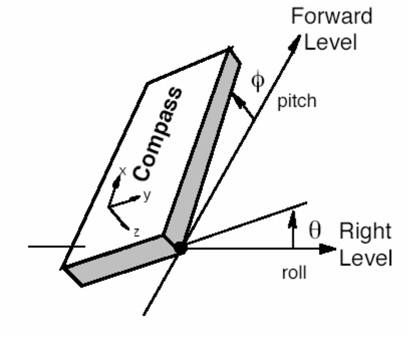


Figure 2.1. Compass tilt referenced to the Earth's horizontal plane

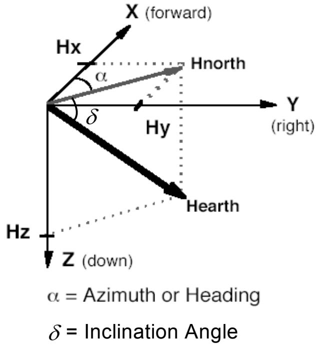


Figure 2.2. Earth's field in X, Y, Z coordinates

$$H_x = b_x \cos\varphi + b_y \sin\varphi \sin\theta + b_z \sin\varphi \cos\theta$$

$$H_y = b_y \cos\theta - b_z \sin\theta$$

where b_x, b_y, and b_z are components measured by the 3-D magnetic compass (Figure 2.1).

The azimuth derived from these values will contain and propagate the errors present in the attitude angles themselves. According to the first order Taylor development of the azimuth computation, the error produced can be written as [2]

$$\Delta\alpha = -\Delta\theta \cdot \tan\delta \cdot \cos\alpha - \Delta\varphi \cdot \tan\delta \cdot \sin\alpha$$

where δ is the angle of the magnetic field to the surface of the earth, also called the dip, or inclination, angle (See Figure 2.2). In the northern hemisphere, the inclination angle is roughly 70° down toward the North.

This relation shows that the error in determining the attitude angles affects directly the azimuth and its effect strongly depends on the azimuth itself. For mid-latitude, the average value of 2 for $\tan\delta$ can be considered.

Errors and Solutions:

Heading error is affected by:

- Sensor platform accuracy (±0.5°): A/D converter resolution, magnetic sensor errors, temperature effects
- Magnetic disturbances: Sources of magnetic fields include permanent magnets, motors, electric currents, and magnetic metals such as steel or iron. For example, a long ferroconcrete building can cause a four-degree heading error (Figure 2.3).
- Compass tilt errors: A one-degree error in tilt can cause nearly three degrees of heading error. (The inclination angle of the earth's field near our office at University of Notre Dame, Indiana, is approximately 70° [3].)
- Variation of the earth's field: The angular difference between the magnetic and true north expressed as an Easterly or Westerly variation. This difference is defined as the variation angle and is dependent on the compass location – sometimes as large as 25°.

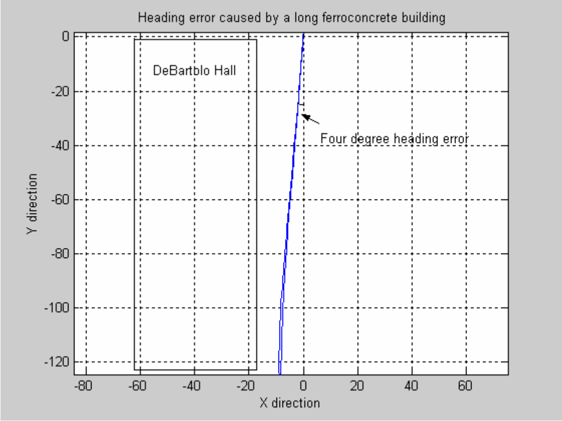


Figure 2.3. Walking close to a ferroconcrete building may introduce a heading error.

Dynamic (tilt) and static calibrations are needed to compensate for compass errors:

- Dynamic calibration: Dynamic calibration will be necessary to compensate the individual errors caused by walking. Low-pass filtering the additional accelerations eliminates the typical oscillations in walking. If the computed pitch and roll values reflect the movements done by the hips, virtual values have to be defined to compensate the displacement effects on the horizontal plane. These additional constant corrections reflect the individual characteristics of a walk and the symmetry between left and right strides.
- Static calibration: When a two-axis (*X*, *Y*) magnetic sensor is rotated in the horizontal plane with no disturbances, the output plot of H_x vs. H_y will form a circle centered at the origin (0, 0). The effect of a magnetic disturbance on the heading will be to distort the circle, resulting in an ellipsoid not centered at the origin. For example, even the human body can offset the circle as shown in Figure 2.4.

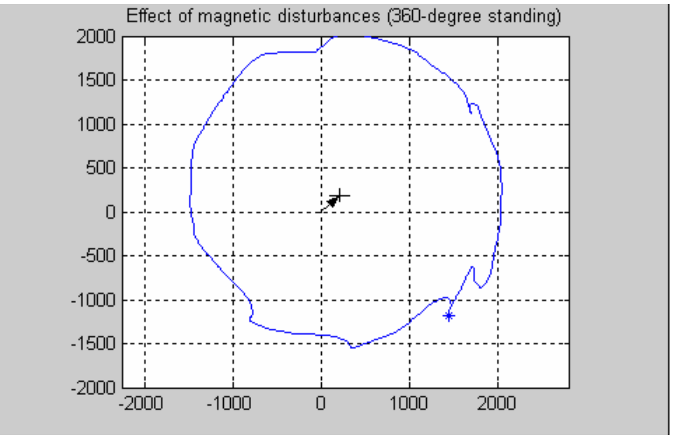


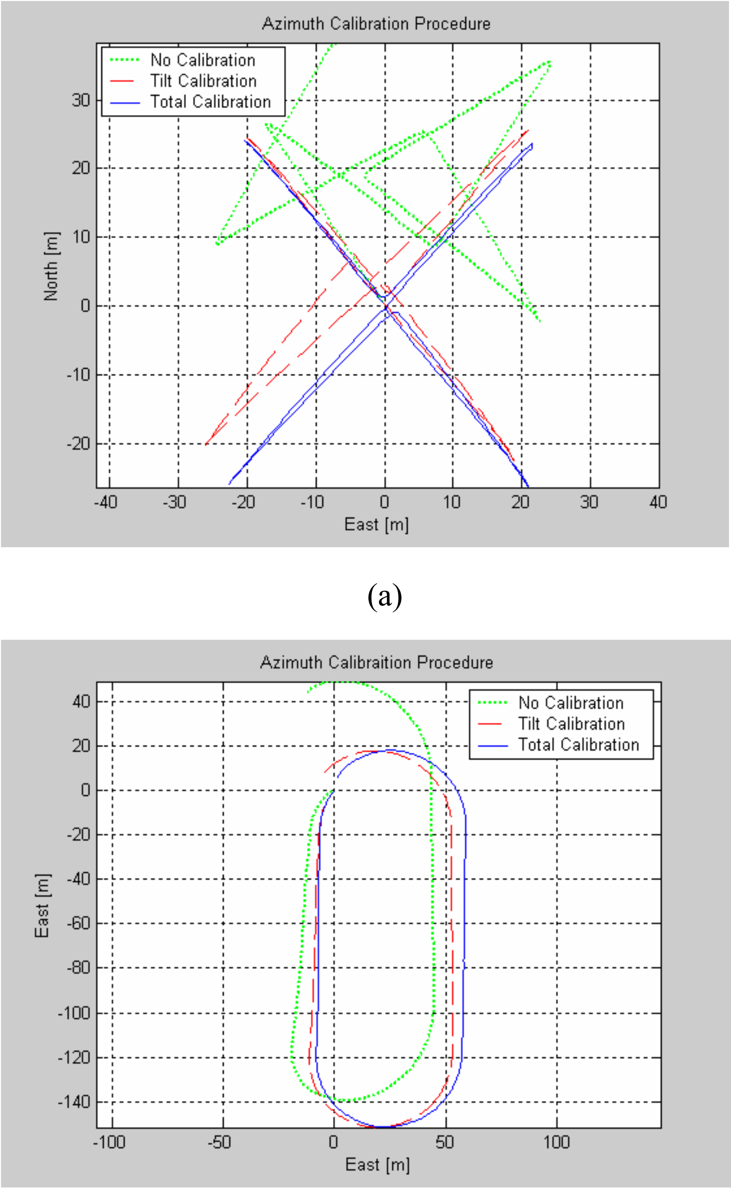
Figure 2.4. The effect of body offset

Two scaling factors X_{sf} and Y_{sf} can be determined to change the ellipsoid response to a circle. Offset values X_{off} and Y_{off} can then be calculated to center the circle close to the origin [4]. Applying the corrections to the projected magnetic values, we have

$$H_x = X_{sf} \cdot H_{x\_mes} + X_{off}$$

$$H_y = Y_{sf} \cdot H_{y\_mes} + Y_{off}$$

The effect of total calibration (dynamic and static calibration) has been shown in Table 2.3. A heading error of 1.1° is obtained after total calibration, compared to a heading error of 6.8°.



(a)

(b)

Figure 2.5. Effects of the different calibration phases. (a) A cross-type walk on DeBartolo Quad at ND; (b) A walk on a 400m track Olympic stadium

2.3. Summary

The achievable precision for the dead reckoning algorithm is:

- Distance estimation: ±%3 with different walking speed on different surfaces
- Azimuth estimation: ±1° in open area; ±5° in urban area (without strong magnetic disturbances)

The performance of the algorithm can be further improved if GPS or gyro information is available.

References

[1] Harvey Weinberg, *Using the ADXL202 in pedometer and personal navigation applications*, Technical Note, Analog Device.
[2] Quentin Ladetto, Josephus van Seeters, Stanley Sokolowski, Zbigniew Sagan, Bertrand Merminod (2002), “Digital Magnetic Compass and Gyroscope for Dismounted Soldier Position & Navigation,” http://topo.epfl.ch/personnes/qlo/Nato_Ladetto_&_al.pdf
[3] Magnetic declination, models, data and services from NGDC, http://www.ngdc.noaa.gov/seg/potfld/geomag.shtml
[4] M. J. Caruso, “Applications of magnetoresistive sensors in navigation systems,” *Sensor and Actuators*, SAE SP-1220: 15-21.

Section 3. Tagmote Hardware

This section describes both electrical and mechanical designs of the Tagmote mobile sensor. Technical drawings and electrical schematics are provided for the reader's reference. Note that here the latest, 2 processor version of the TagMote is described. Details of earlier versions may be found in the PI presentations that accompany this report.

3.1. Electronic Design

The latest Tagmote design consists of four main components the MICA, the slave processor board, the interface board, and the Leica sensors. Figure 3.1 displays a block diagram of the Tagmote. The slave processor is in charge of executing most of the data processing while a Berkeley MICA platform handles the communication and networking activities. The Leica Geosystem's DMC-SX provides a digital magnetic compass and a three-axis accelerometer, which is connected to the Leica through a specialized board that contains a 4MB FLASH memory for data storage.

The TagMote uses a Berkeley MICA platform [1] (reference numbers refer to the list at the end of the present section) for general coordination and communication. Its central processor is an Atmel ATMEGA128L 8-bit processor running at 7.3728 MHz. This microcontroller has 128 KB of programmable flash memory and 512KB of external FLASH memory. The MICA also provides radio communication at 433 MHz (Note that an earlier version of MICA used 916 MHz for radio communication; the current version with 433 MHz is designated as MICA2). The radio has 8 channels and is capable of transmitting data at a rate of 38.4 Kbaud with a range of 1000ft (the radio transmission range is severely affected by ground absorption effects). A 51-pin connector is used to interface with the MICA platform.

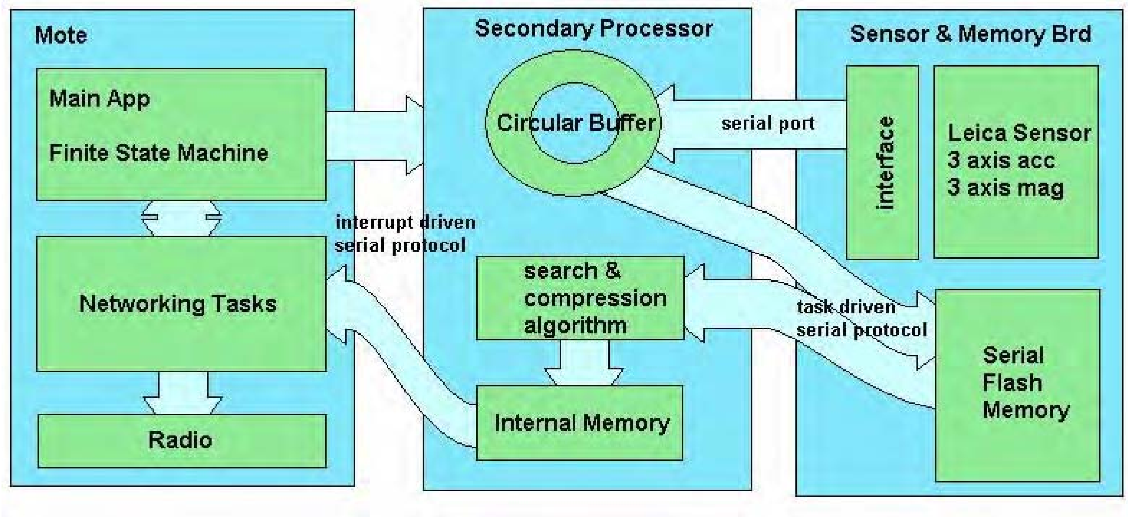


Figure 3.1 Tagmote block diagram.

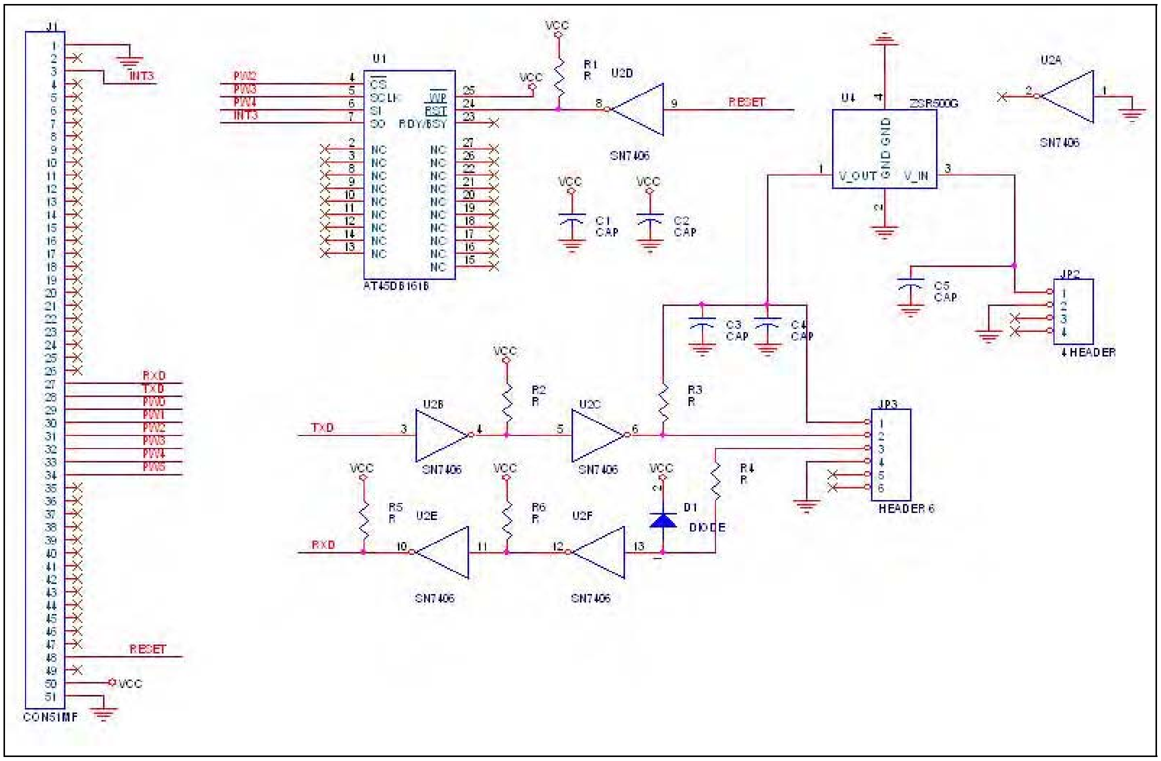


Figure 3.2 Schematic of Tagmote interface board.

The Leica sensor was selected because of its sensor capability, small size, and the ability for data filtering. The Leica has a 3-axis magnetometer and 3-axis accelerometer with data rates up to 50 Hz [2]. The size of the Leica is 33 mm x 31 mm x 13.5 mm. It has flexible software with 13 operating modes, which provide different levels of compensation. Communication with the Leica is done by means of one serial interface, which is protected against non-valid commands.

The interface board connects the Leica to the slave processor board and provides 4 MB of additional Flash memory for data storage. A schematic of the Tagmote interface board is given in Figure 3.2. The Leica sensor connects to JP1, which supplies power (5V), ground, serial data input and output.

The slave processor is in charge of configuring the sensor, retrieving measurements from it, pre-processing the information, storing them into the flash memory, and handing them out to the MICA processor to be transmitted over the network. The slave processor board is augmented with its own ATMEL128L processor that draws its power from the 51 pin connector to the MICA processor. The slave processor board interfaces to the interface board using a 51 pin connector compatible with the MICA processor. The communication between the MICA processor and the slave processor is done using a task driven custom serial protocol that uses interrupts as acknowledgements.

The input and output signals for the Leica sensor are driven by the TXD and RXD pins of the slave processor serial port. The memory chip is the AT45DB161B U1, this serial flash data chip has a serial interface and works with 3V, which makes it ideal to connect to the MICA where small interface pin count and low voltage operation is required. The reset pin was left unconnected since the chip provided its own reset circuit.

Open collector inverters, 74HCT06, provide the voltage translation to connect the 5V Leica sensor logic to the 3V slave processor logic. These inverters are labeled U2B, U2C, U2E, U2F in Figure 3.2. Inverters are used instead of a buffer because we want to connect the slave processor reset pin to the memory, which already required an inverter. Open collector inverters have at their output a transistor with nothing connected at its collector end and grounded at its emitter. The collector end is available at the output pin. So this can be used to transform the logic at their input (which in this case is 3V) to any higher voltage (5V) by placing a resistor connected at this higher voltage. When the output is low, the transistor is saturated and a voltage of about 0.15 Volts can be obtained with the appropriate resistor (R3). When the output is high, the transistor is off and the voltage on the output will be approximately 5V is the input being fed has much higher impedance to ground than the resistor used. On the other hand U2F is used to translate the 5V output logic of the Leica sensor to the 3V input logic of the slave processor. Here D1 is used to prevent the input to U2F be higher that 3.5V while R4 is used to limit the current that goes through D1 when the output of the Leica sensor is 5V. All other resistors are needed to place a high level of 3V to the inverters outputs.

All capacitors used are decoupling capacitors for the 3V, 5V, and unregulated power supply for the linear 5V regulator U4 ZSR500G. The unregulated power supply is fed to the circuit though JP2. For our tests, we used a 9V alkaline battery to provide the unregulated power. Note that, relatively large decoupling capacitors are needed for the 5V power regulator (>20uF) since the Leica draws large intermittent amounts of current during its power-up.

Atmel provides several memory chips with the same footprint and same interface, which allows future memory expansions by just replacing chip U1. We are currently using an Atmel DATAFLASH AT45DB161B 3V read/write FLASH memory chip, which provides 4MB of data storage [3]. The interface consists of SCLK (serial clock), SO (serial output), SI (serial input), and CS (chip select). All pins are inputs except SO. All transactions are started by selecting the chip with CS. After that data can be shifted in through SI or out through SO, each bit is clocked using SCLK which is driven by the MICA. The protocol that manipulates the pins appropriately was implemented by software.

3.2. Mechanical Design

Several different housing cases were developed to protect the TagMote's electrical hardware and sensors from environmental damage. Some designs housed both the batteries and Tagmote together. However, the current design separates the power supply from the MICA and Leica sensor. This was done for a couple of reasons. First, designs that held the circuitry, sensors, and power supply were much larger and had a tendency to cantilever during testing, which produced significant error in the data. Furthermore, the close proximity of the batteries to the Leica sensor tends to cause a large disturbance in the magnetometer readings.

The current design consists of an epoxy case, which houses the MICA, interface board, and Leica to prevent them from environmental damage, and a hand-held case for the power supply. Technical drawings of the Tagmote case are provided at the end of this section. Note that this case was used for the second version of the TagMote and photos may be found in the PI presentations that accompany this report. Stereolithagraphy was used to manufacture the housing case because it provided the flexibility needed for the design and was very economical.

Reference

[1] Crossbow Technologies, “MICA 2 Datasheet”, available at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-01_A_MICA2.pdf, 2003.
[2] Leica, *DMC-SX Digital Magnetic Compass User Manual,* Version 3.5, Release 6, February 2002.
[3] Atmel, *AT45DB041B Datasheet*, January 2001.

Tagmote Hardware Characteristics

Processing MICA2 board

Processor:	Atmel ATMEGA128L 8bit RISC @ 7.3728MHZ
Program Memory:	128 Kbytes
Program RAM:	512 Kbytes
Radio:	38.4 Kbits/sec @ 433 MHz
Operating Voltage:	3 Volts

Slave Processor board

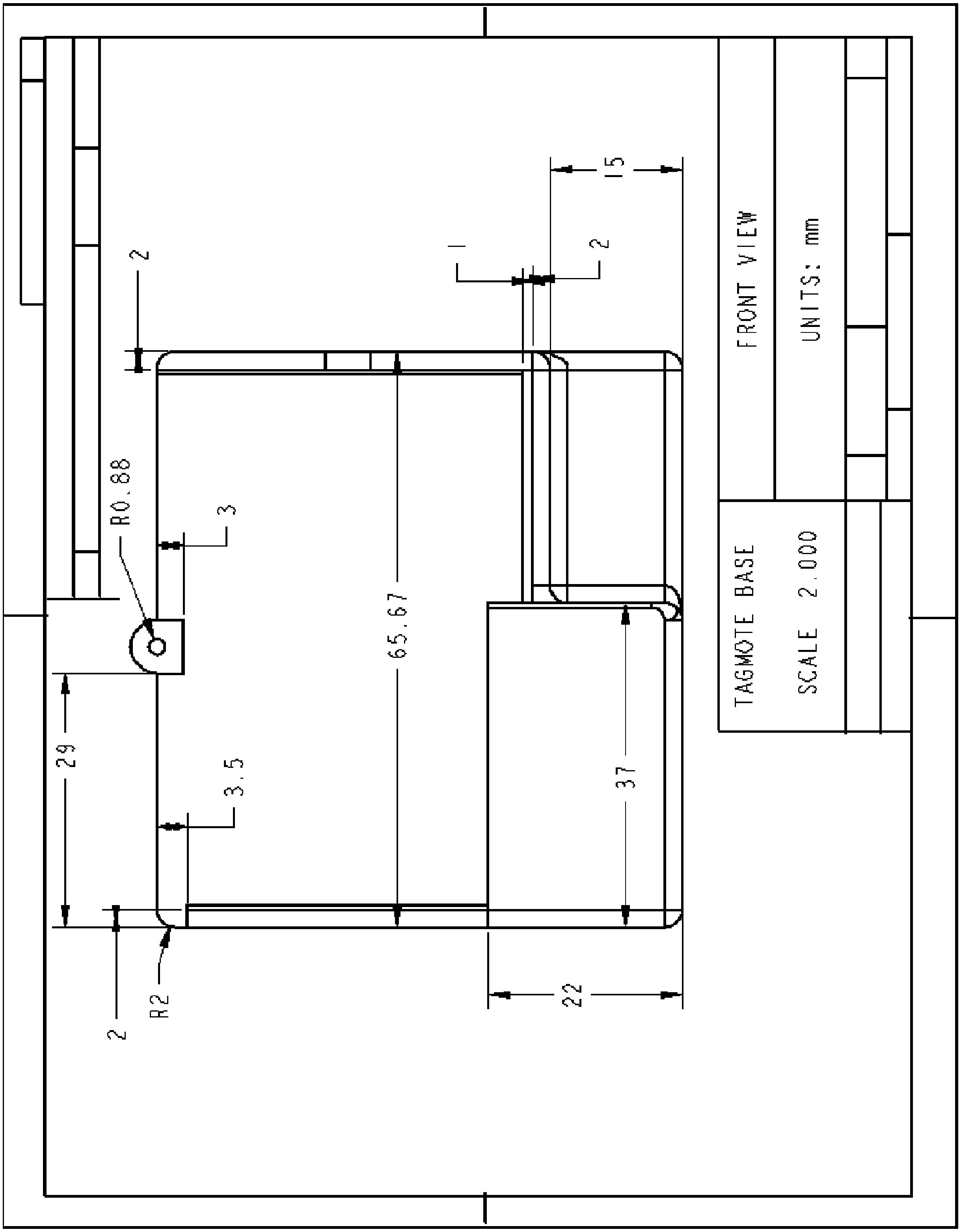
Processor:	Atmel ATMEGA128L 8bit RISC @ 7.3728MHZ
Program Memory:	128 Kbytes
Program RAM:	512 Kbytes
Radio:	none
Operating Voltage:	3 Volts

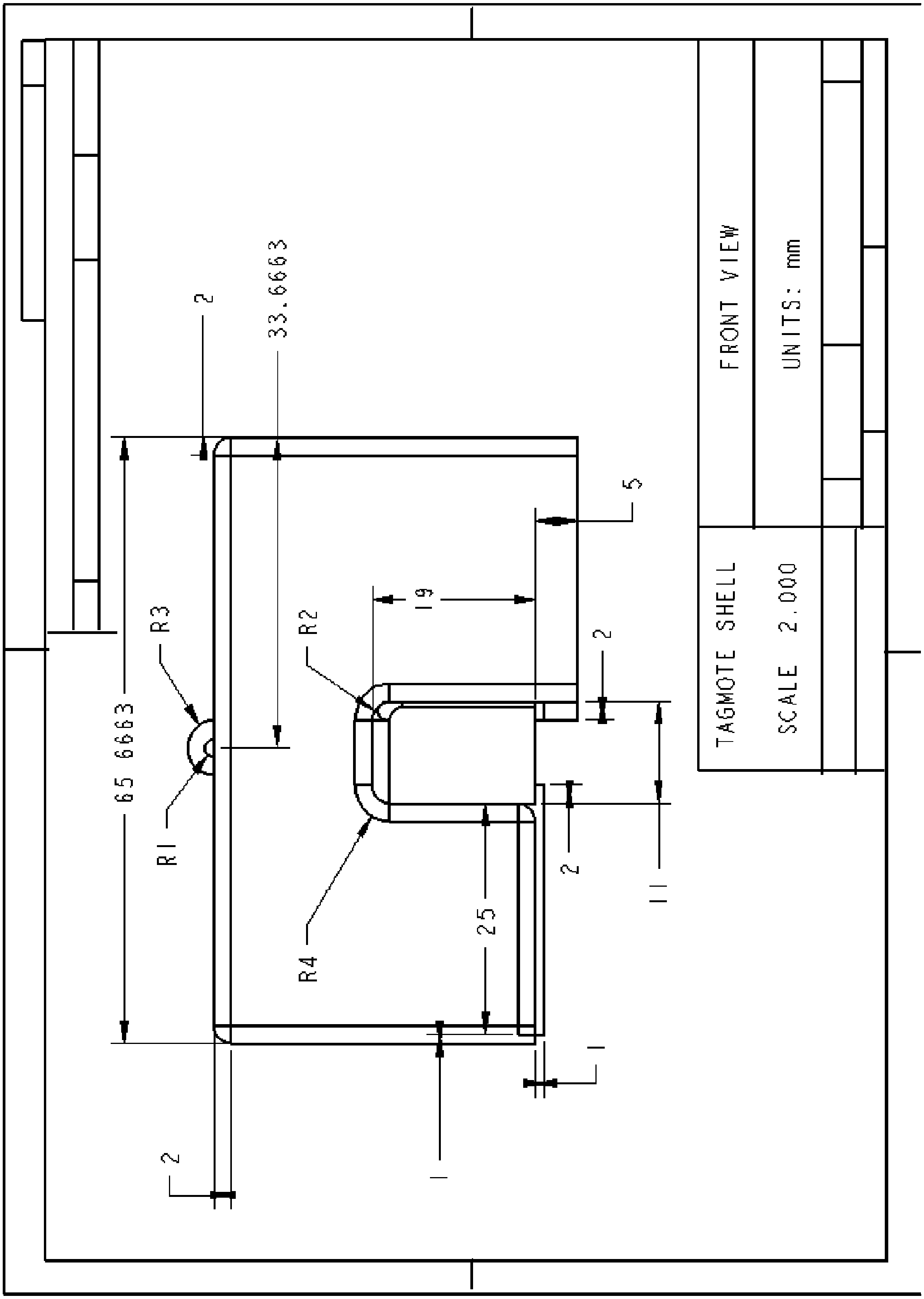
Sensor & Log Memory Board

Sensor:	Leica 3 axis accelerometer 3 axis magnetometer
Interface:	UART w. voltage translator @ 9600 bits/sec
Sampling Freq:	30 Hz
Data Frame Size:	28 bytes/sample
Effective Data Size:	12 bytes/sample
Sensor Voltage:	5 Volts

Memory Chip:	Atmel Data Flash AT45DB161B
Memory Type:	Serial Interface Flash Memory
Memory Capacity:	16 Mbits (4MB)
Maximum No Samples:	variable: aprox 12 hours
Memory Voltage:	3 Volts
Maximum Operating Freq:	20 MHz
Interface:	Interrupt-driven RAM circular buffer data collection, Task-driven write/read software serial interface.

Technical Drawings of Tagmote Case




TAGMOTE SHELL
SCALE 2.000
FRONT VIEW
UNITS: mm
65 6663
33.6663
2
5
19
2
R3
R2
R1
R4
25
2
11
1
1
2

Section 4. Tagmote Software, NEST Middleware and Base Station Software

Recall the setting of the Red Force Tagging project. A ***Tagmote*** acquires data, and then downloads it to a network, when the Tagmote is in the proximity of the network. The network consists of a number of motes, which we call ***network motes***. The ***RelayMote*** is a special network mote which is physically connected to the ***base station***. The data downloaded to the network is sent to the RelayMote, which forwards it to the base station. The base station processes the data and displays it. All communication is wireless, except for the communication between the RelayMote and the base station.

The software developed in this project can be divided in several categories:

- Tagmote specific software
- NEST Middleware for
    - The network motes
    - The RelayMote
    - The Tagmote
- Base station software for
    - Network monitoring/debugging
    - Data decoding
    - Data processing and display

This section contains a description these software components, except for the display software, which is detailed in the next section. The software exists in several working versions. The latest version of the NEST middleware allows for data compression.

A brief description of the main functions of the software is as follows.

- The Tagmote software acquires data from the Tagmote sensors, processes it, and saves it in the memory. The processing includes a form of data compression, which increases the amount of data that can be stored in the memory and decreases the download time. After the prescribed amount of data is recorded in the memory, the Tagmote begins searching for a network. When a network is found, the data is downloaded to the network. The Tagmote software is distributed on two processors, containing also a software module governing the communication between the processors.
- The middleware software provides networking services, including a global time (synchronization), routing services, and data logging. Thus, the network motes that receive data from the Tagmote can save it by means of the data logging service, and then route it towards the relay by means of the routing services.
- The base-station software receives the data forwarded by the relay, and decodes (decompresses) it. The decompressed data is then sent to a java tool that processes it and displays the calculated trajectory graphically.

4.1. Tagmote Software

In addition to some of the NEST middleware components, described in the next subsection, the Tagmote includes:

- “hardware” components (that is, components that monitor the hardware)
- a communication component

The communication component allows a distributed implementation of the software on two processors. One processor is on a regular mica2 board, while the other on a custom made board. A master/slave like architecture is used in which the mica2 board is the master and the custom made board the slave. The master runs the middleware software and the communication component. The slave runs the hardware components and the communication component.

The communication between processors uses one line for serial communication and two lines for interrupt signals. The communication is asynchronous. To avoid slowing down more important tasks, communication is implemented by a low priority task. Thus, the data rate depends on how busy the processors are. The communication uses a protocol that ensures data is transferred in a reliable manner; only hardware failures may cause transmission errors.

The “hardware” components also implement some real-time processing of the sensor data. The role of this processing is to reduce the download time and filter the sensor data. The download time is reduced in two ways:

- Some of the processing required to estimate the trajectory from the raw data takes place in the Tagmote, in real-time. Thus, the Tagmote does a min/max detection, required by the estimation algorithm. Since only the min/max values are sent, 8 times less data (on the average) is sent to the base station.
- Data is compressed, by using differential measurements. A differential measurement consists of the difference between the current data and the previous data. If data does not vary very fast, fewer bits can be used for the difference than for the current data value. On the average, this technique speeds up the download time by a factor of 2.

The Tagmote “hardware” components are:

- HWTAGMOTE: This component is in charge of orchestrating all the hardware components and providing an interface for the top-level application. Real-time processing of sensor data is also done in this component. Further, the component initializes the lower level components, initiates the process of data acquisition and storage of the data received from the Leica sensor to the Flash memory, stops the acquisition of data when the number of lines read reaches a certain number, and finally it issues read commands to the flash that will trigger a data read ready event to retrieve the data from the Flash.
- FLASH: This component manages the Serial Flash chip. The commands that can be issued to this component are: Start Reading, Start and Stop Writing, Write, and Read. The actual processes that involve directly the Flash are managed using tasks. This avoids blocking the microcontroller from performing other interrupt driven tasks. The Write command stores the data in a circular buffer (100 bytes). A task called WritingProcess is in charge of transferring the data in the circular buffer to the Flash memory. The circular buffer structure allows a continuous generation of data to be transferred to the Flash in bursts. This is because the

Flash data is written first to an internal buffer; after the buffer is full a command is issued to transfer the contents to the main memory. The WritingProcess task keeps track of the internal buffer status and initiates the transfer to main memory. It will repost itself after each action, until no more command to the Flash are to be issued and the temporal buffer is empty. The Read command posts a ReadingProcess task. Similarly to the WritingProcess the data is read through a Flash internal buffer and not directly from main memory. Therefore ReadingProcess keeps track of the contents of the internal buffer and initiates the transfer from main memory to the internal buffer when necessary. After a data byte is retrieved, ReadingProcess generates an event with the read byte. Since the read and write commands set the Flash memory in different modes a complex system of semaphores and flags is used to prevent both commands occurring at the same time.

- MODEB: This component generates the appropriate command sequences for the Leica sensor that sets the Leica in continuous measurements of three accelerometers and three magnetometers. It also decodes the data sent by the Leica sensor.
- UARTPACKET: It receives the data to be sent back and forth to the Leica Sensor. It receives data from the Leica in a byte-by-byte fashion, strips the frame from its overhead and forwards the payload data to the MODEB component. It also receives the frames to be sent to the Leica and transmits them byte by byte to the sensor through the UART component.
- UART: It's the component that manages the UART hardware, it initializes the hardware and receives and transmits the bytes from and to the UART.

The structure of the hardware components is shown next:

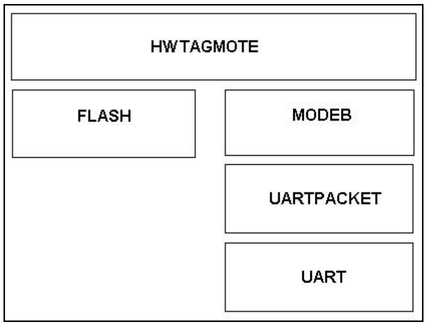


Figure 4.1 Hardware components structure

The operation of the master is as follows. The master waits for the slave to signal that it has finished acquiring sensor data. Then, it begins to search for the mote network, in order to download the data. The search is done by means of the ping component of the middleware. The PING component sends out Hello messages until a network mote answers indicating that it is ready to receive the data from the Tagmote. Once the PING component receives such an acknowledgement, it generates an event in TAGMOTE that in return issues a dump command. The DUMP component requests data from the slave, and then forwards it to the pinged mote. For every transmitted frame, the receiving network mote replies with an acknowledgement message. If an acknowledgement message is not received, the DUMP process is stopped and the PING process is started again to find a new recipient for the data. The download process is complete when the slave signals it has no more data to send.

The Tagmote code has several working versions. The older versions moved gradually from a centralized (one processor) implementation that was simply sending raw data to the network, towards a distributed (two processor) implementation, in which data undergoes some real-time processing, including a data compression process. Other technical improvements were also added along the way.

4.2. NEST Middleware

The NEST middleware consists of a number of services for network communication. Each of the Tagmote, RelayMote, and network motes uses some of the middleware services, as described below.

The **broker service** builds a spanning tree in which the relay is the root and the network motes appear as nodes or leaves. The network motes use this service in order to determine the next hop, when they forward the data downloaded by the Tagmote towards the relay. The broker service relies on the synchronization service. The broker service is used by the network motes and the RelayMote.

The **dump service** is the name of the Tagmote service that sends data to a network mote and also the name of the service of the network motes that stores locally the data transmitted by the Tagmote. This is the protocol: The Tagmote sends ping signals periodically. When the Tagmote is in the range of the network of motes, the motes that hear the ping signals reply. Then, the Tagmote establishes the connection with one of the network motes that has replied and dumps data to that mote until there is no more data to send or no acknowledgments to the data sent are received. The latter may happen when (i) the Tagmote position has changed and no longer allows communication with the network mote or (ii) the network mote has filled its storage space with data. In either case, the Tagmote will start again sending ping signals. The network motes store the data dumped by the Tagmote in the local EEPROM.

The **neighborhood service** maintains a list of the (active) neighbors a mote has. Note that mote B is a neighbor of mote A if A and B can communicate directly (i.e. single hop). Note that the operation of the synchronization service relies on the neighborhood service. The neighborhood service is used by the network and RelayMotes.

The **ping service** is the service used by the Tagmote when it seeks to connect to the network. The same name is used to denote the network mote service that replies to the ping messages sent by the Tagmote. The operation of this service is as follows. The Tagmote sends ping messages. If a network mote hears a ping message, it replies with a message that includes its own address. Then, if the Tagmote hears any reply, it selects one of the replying motes as destination for the dump service.

The **publication service** transmits the data stored in the network motes to the RelayMote. Each mote that has stored data dumped by the Tagmote, starts (at a random time) forwarding data packets toward the RelayMote. The protocol that is used is best seen in the source code (PublishM.nc). The protocol associates a buffer to each mote. The buffer is filled either from the EEPROM, if the Tagmote has dumped data there, or from data received from other motes. Note that data is progressively sent toward to the RelayMote according to the routes described in the spanning tree produced by the broker service. A mote X starts (at a random time) sending the data from its buffer toward a neighbor mote Y by means of a ``set'' message. Y acknowledges with an ``ack'' message, and then X starts sending data packets with ``dat'' messages. Y replies with ``ack'' messages. Either of X and Y can clear (close) the connection with a ``clr'' message. There are also locking and timeout mechanisms that ensure the proper operation of the protocol. The service achieves about 400, 250, and 100 packets/min for one, two, and three hops, respectively. (The throughput goes down as we increase the number of hops in the route.) Note that the relay version of the service forwards to the base station the packets received from the publication service of the network motes.

The **route service** provides means of detecting the state of the network. Thus, the network motes use this service to send to the relay data packets that contain the local address (i.e. the identity) of the mote and the local addresses of its neighbors. The relay forwards the packets it receives to the base station.

The **synchronization service** is used by the network and RelayMotes to create a global clock. In this implementation, the clock has a period of roughly 40sec. The synchronization service is implemented together with the neighborhood service in HoodM.nc. Experimental results indicate a variance of 10msec in event synchronization.

The **telemetry service** is useful for debugging purposes. It implements commands that are received by radio. This service has been used in a setup involving a computer running the NestConsole java tool that is also connected to a GenericBase mote (i.e. a mote that runs the GenericBase application of the TinyOS 1.x distribution). In this setup, the user can send commands to the network motes using the NestConsole tool. The GenericBase mote is used to forward these commands and to listen to the network messages.

The middleware services have several working versions. The relay and network mote version that can handle Tagmote packets with compressed data is called NestRelay_verC.1, consisting of wNestMica2_ver2.8C (the extension for compressed data of the 2.8 middleware version of the network motes) and wRelayMica2_ver2.1C (the extension for compressed data of the 2.1 middleware version of the RelayMotes). Note that the compression version and the previous versions are incompatible: the compression version of the middleware cannot handle the data format of the software without compression, and the version of the middleware for uncompressed data cannot handle compressed data. Thus, the latest version of the Tagmote software can only be used with the NestRelay_verC.1 code, and the previous versions of the Tagmote software cannot be used with the NestRelay_verC.1 code.

4.3. Base Station Software

The base-station software consists of java tools and the executable *undif,* implementing a C program. A java tool, called NestConsole is responsible of saving the data packets received from the RelayMote in a file. That file is then read by undif, which decompresses the data. Finally, the decompressed data is processed by a java tool for processing and displaying the data.

Note that there are two forms of the NestConsole tool. One, which is found with the network mote software, is meant to work with a GenericBase mote, instead of a relay. The other one, found with the RelayMote software, assumes the base station is connected to the relay. Both NestConsole tools plot also the status of the network. For instance, links are used to display the motes that are neighbors.

There are also several versions of the NestConsole tool. The older versions have been used with the middleware for uncompressed data. The latest version is a slight modification that allows proper interfacing with the decompression tool undif.

4.4. Outline of System Operation

1. From the Tagmote to the network motes:
When the Tagmote is ready to dump data, it pings to connect to the network. If a network mote can hear the ping message and has an active parent to which it can transmit the data, it will respond to the ping. Then, the Tagmote begins to dump data to the network mote. When the data buffer of the network mote is full, the Tagmote looks for another network mote and continues downloading its data.

2. From the network motes to the relay:
When a network mote has data to transmit to the relay, it chooses its parent in its routing list and tries to transmit. If the trial fails (no acknowledgements are received), the network mote will change to another parent in its routing list until it is successful. Then the network mote begins transmitting data to its parent. This is repeated until the data reaches the relay. Note that if the network mote loses the link in the middle of a transmission, it has the ability to change its parent and set up a new path to communicate.

4.5. Files

Note that the software code developed for this project has been delivered to DARPA and is not included with this report.

The most recent software is included in the following files:

- The middleware for the relay and network motes as well as the java tool NestConsole and undif are found in NestRelay_verC.1.zip. Included are also examples of installation and data processing shell scripts (inst and traj). Note that the middleware requires TinyOS 1.0. Some minor updates may be required to compile it in the more recent versions of TinyOS.

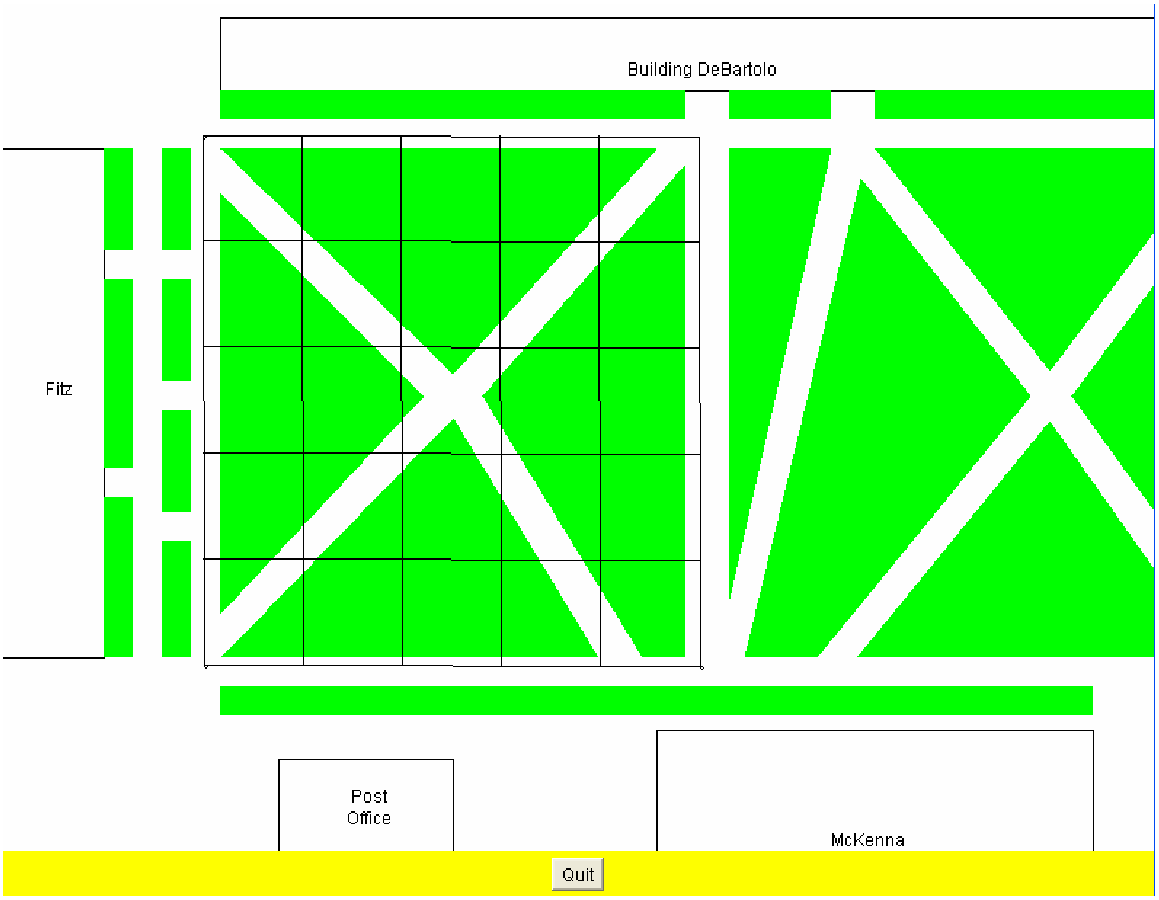
- The Tagmote software is found in MR5.1.zip. Note that it consists of two components: the code for the master processor (the one on the mica2) and the code for the slave processor (the one on the mica radio board). Note also that the code requires TinyOS 1.1.

Section 5. Display and GUI Design

This section explains the steps taken to process and display the data when they are received by the Relay mote.

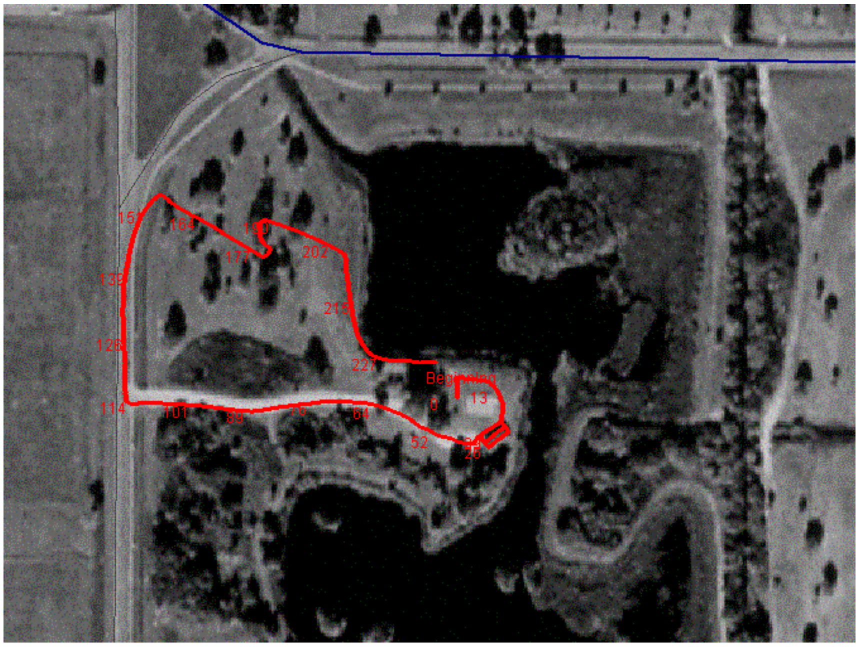
5.1. GUI Design Using Java

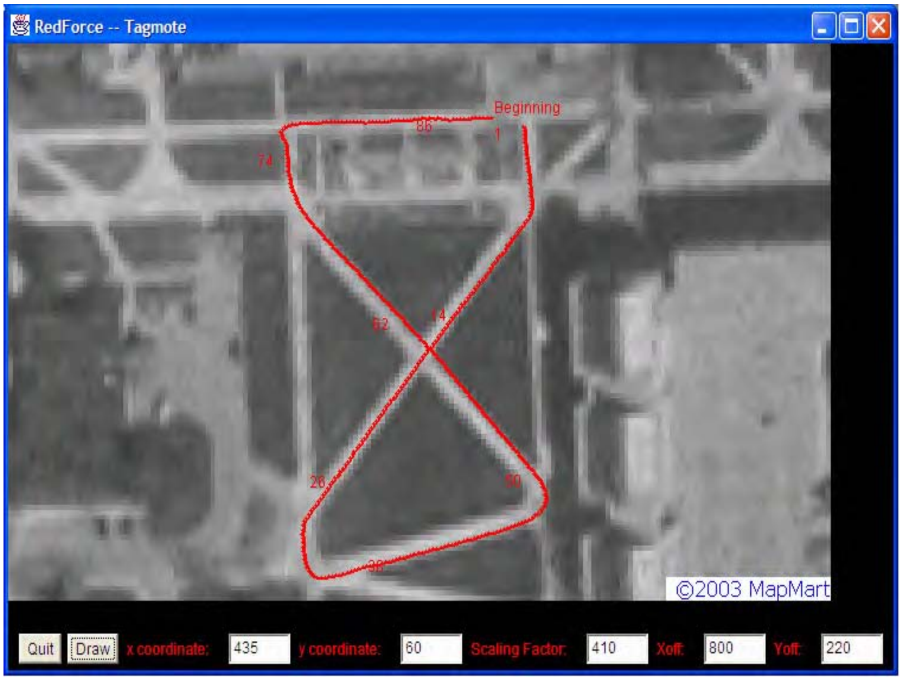
The first attempt was to create an applet, so that the application could be executed and monitored online. The terrain was manually drawn, using simple graph methods (JAVA). The points were being read from a file. The interface was as shown below.



The second step was to create a stand alone program that would use the canvas and frame routine, so that it would be platform independent and would be executable on all machines. The program was first reading the contents of a text file, containing 3 columns, separated by commas, that is x, y, time and depicting them on a map (either an icon or a graph). Later on the code was changed from reading a text file to listening to the standard input for data in the same format. At this point the code was also made more independent, regarding the length and format of the data, the only restriction was that the 3 columns should be separated by commas. This version had a big disadvantage. If we wanted to change the starting point or the scaling factor of the trajectory (the latter is associated with the K factor, which depends on the subject), we had to change the code and recompile. This problem was solved by adding text boxes on the interface, two for the coordinates of the starting point, one for the scaling factor and later on, two for the xoff and yoff parameters (to adjust the Leica module). So the user could insert the appropriate values in the boxes and redraw the trajectory to make it match closer the terrain.

The final interface looks as follows:





x coordinate: 435 y coordinate: 60 Scaling Factor: 410 Xoff: 800 Yoff: 220

5.2. Implementation of DR Algorithm Using Java

After the Dead Reckoning (DR) algorithm was developed and fully tested, the data that arrived at the Relay were exfiltrated and the algorithm was implemented using Java for easy system integration.

Since Java lacks the predefined functions for matrix and vector computation, those functions are first coded. With the prototype algorithm in Matlab, it was relatively easy to debug the Java program. Using the same raw data, the results from the Java program matched well the results from the Matlab program. (Difference was less than 0.001). The Java code was shown to be robust and accurate. This Java program involves two versions: the first version corresponded to our initial design, the board where the raw data from the sensors were used as our input data. The second version corresponded to a later design, when we had added the preprocessing capability to the Tagmote. At the Relay we were processing only the min and max data points extracted by the preprocessing algorithm. With the well-designed class-structure and functions of the first version, the second version was only a minor simplification of the previous version, which justified the notion of software reuse. The output from this program was set of points of the walked trajectory.

5.3. Visualization of the Trajectory

After computing the trajectory points, the next step involved plotting the trajectory for visualization. The terrain JPG file was loaded as the background and the trajectory was plotted against this background. In a typical situation, initially the scaling and orientation of our trajectory did not match the actual path, since we did not know the ratio of the physical distance and the screen distance. This problem was solved by introducing a scaling factor that could be easily adjusted on the screen, so that a good match could be obtained. Considering the compass body offset, we used another two parameters to adjust the offset of the magnetic readings. After the 1st demo, we integrated the DR algorithm and visualization as one program instead of using a “pipe” command to put them together. The ability to use the mouse to choose the starting point was also added.

DARPA IXO/NEST Award No. F30602-01-2-0526

Final Report

Part II

Analytical Modeling, Performance Evaluation, and Optimization of Large Systems of Networked Embedded Sensors

Section 1. Executive Summary

Analytical Modeling, Performance Evaluation, and Optimization of Large Systems of Networked Embedded Sensors

Networked Embedded Wireless Systems have a dynamic multihop topology composed of power- and bandwidth-constrained wireless links. The topology changes with time as the qualities of the wireless channels vary and as the nodes move or adjust their transmission and reception parameters; it cannot be determined or predicted and therefore has to be modeled as random. Therefore our objectives of this theoretical part were
(1) to model the topological uncertainties of large sensor networks and
(2) to identify solultions to cope with and manage the inherent non-determinism.

Related to network topologies, we derived general analytical expressions for the internode distances and showed that nearest-neighbor routing schemes perform poorly in terms of energy consumption and throughput. For better performance, we introduced quasi-regular networks with significantly smaller distance variance and much improved coverage properties. At the link level, we introduced simple yet accurate analytical packet reception models that incorporate fading as a small-scale variation in the received signal power. This model was shown to predict the packet reception probabilities observed in real networks rather well. We also studied the throughput performance of interference-limited networks and extended standard "disk graph" models to graphs with non-isotropic connectivity. Further investigations included load balancing problems, queueing-theoretic delay analyses, the impact of power amplifier characteristics, and opportunistic channel access and routing schemes.

Section 2. Description of Research Results

1 Motivation

Due to the inherent non-determinism and computational and communicational limitations, networks of embedded systems require sound mathematical models that capture the probabilistic and dynamic character of these networks, including the random node distribution and the unknown channel gains and interferences. Of crucial importance is the ability to predict, analyze, and bound performance for large systems. This part of the final report lists our contributions to this problem.

2 Research Results

The emphasis was put on (1) modeling the topological uncertainties of sensor networks and (2) finding solutions to cope with and manage the inherent non-determinism. At the topology level, there are mainly two sources of uncertainty: the node distribution and the characteristics of the wireless channel.

2.1 Node distribution and internode distances

Due to the rapid decay of signal power levels with distance in RF communications, uncertainty in the location of the nodes results in a compounded uncertainty in the SINR. Although perfectly regular lattices may be useful assumptions to derive (upper) performance bounds, in practice, the node distribution is usually subject to some uncertainty.

Poisson random networks. A well accepted model for the node distribution[1] is the homogeneous *Poisson point process* of intensity λ, where the number of nodes in an area A is Poisson distributed with parameter $A\lambda$ and the node numbers in disjoint areas are independent. We derived the following general result on the distances in Poisson random networks [1]: For an m-dimensional network, the distance R_n between a node and its n-th neighbor has the probability density function (pdf)

$$f_{R_n}(r) = e^{-\lambda c_m r^m} \frac{m\,(\lambda c_m r^m)^n}{r(n-1)!}\,, \tag{1}$$

where $c_m r^m$ is the volume of the m-sphere of radius r. Assuming a path loss exponent of α, the (normalized) mean energy consumption $\mathbb{E}[R_n^\alpha]$ (for n-th nearest neighbor routing) can be expressed in closed form [2]. Rather than mean values, maxima may be more relevant, in particular for route or network lifetime. For a nearest-neighbor route with M hops, we proved that the expected maximum energy consumption grows at least as $(\ln M)^{\alpha/m}$. This reveals one of the problems of nearest-neighbor routing in Poisson random networks and indicates that other routing strategies or node distributions are preferable.

Based on these finding, we addressed the routing problem in large Poisson random networks in [2] and compared the throughput performance of regular and random topologies in [3]. The main conclusion is that networks with random node distributions suffer from a substantially smaller end-to-end throughput.

[1] In particular, if nodes move around randomly and independently, or if sensor nodes are deployed from an airplane in large quantities.

Quasi-regular sensor networks. Clearly, more regular distribution may be desirable. In [4] we introduced a model where nodes are placed approximately in a regular grid, with the deviation from the ideal grid point being Gaussian in both coordinates, *i.e.*, Δx, $\Delta y \sim \mathcal{N}(0, \sigma^2)$, such that σ controls the regularity of the grid. The distance of a node from its grid point is Rayleigh distributed with mean $\sigma\sqrt{\pi/2}$, and the nearest-neighbor distance is Ricean. The same quasi-regular distribution is obtained by the following thinning process: Starting with a Poisson random network of intensity $\lambda > 1$, pick for every integer lattice point the node closest to it. It is interesting to note that the distance of a node from its grid point is Rayleigh, as it is for the Gaussian quasi-regular network. This strategy balances the energy consumption and the coverage by turning Rayleigh distances into Ricean distances. The drawback is that a higher number of nodes is required to achieve good regularity. On the other hand, by slightly shifting the lattice from time to time, a lifetime benefit is achievable.

2.2 Wireless channel models

The purpose of a channel or link model is to yield a packet reception probability p_r given the link distance d and the other relevant network parameters.

A link model with fading. Two channel or link models are prevalent: the *disk* or *protocol model* and the so-called *physical model*. Both are entirely deterministic and have severe shortcomings. Since most networks, exhibit some type of fading, *i.e.*, a stochastic variation in the received signal power that may be caused by multipath propagation, scattering, or obstruction, we introduced a tractable yet accurate packet-level link model in [2]. This model was incorporated into the "Prowler" simulator that was developed at Vanderbilt University, also as part of the NEST project. In this *Rayleigh fading* model, the packet reception probability $p_r = \mathbb{E}_I[\mathbb{P}[\gamma \geqslant \Theta \mid I]]$ can be factorized into the reception probability of a zero-interference network p_r^N and the reception probability of a zero-noise network p_r^I as follows:

$$p_r = \underbrace{\exp\Big(-\frac{\Theta N}{P_0 d_0^{-\alpha}}\Big)}_{p_r^N} \cdot \underbrace{\prod_{i=1}^{k} \frac{1}{1 + \Theta \frac{P_i}{P_0}\big(\frac{d_0}{d_i}\big)^{\alpha}}}_{p_r^I}, \qquad (2)$$

where k denotes the number of interferers, d_0 is the distance of the desired transmission, and d_i the distances of the interferers.

Interference and throughput. Interference is critical in large-scale sensor networks, since it is the limiting factor for the scalability. Except for the case where the interferers all have the same distance from the receiver, little is known about the cumulated interference. We inter- and extrapolated the few available results, characterizing the interference and the local throughput for a broad class of networks, specifically one- and two-dimensional networks with random and deterministic node placement (Poisson or deterministic/regular), channel access (slotted ALOHA and TDMA), and channel characteristics (deterministic and Rayleigh fading) [5]. Thus we captured up to three sources of non-determinism: the node positions, the channel gains, and the set of simultaneously transmitting (interfering) nodes (dictated by the MAC scheme).

Graphs with non-isotropic connectivity. It was conjectured that "the conditions for connectivity and coverage are insensitive to the shape of the region that a Poisson point covers". This conjecture was recently disproved, and it is shown that, among all shapes with the same area, the disk is the last to lead to percolation, *i.e.*, the disk graph needs the highest expected node degree to percolate. A related, rather surprising result is that transmission to a randomly oriented sector of angle δ and radius r leads to percolation as soon as each node can, on average, reach $1 + o(1)$ neighbors. This result suggests the use of directional transceivers, but the problem of isolated nodes needs to be avoided[2], since the gap between percolation and connectivity is accentuated. So, the source and destination node may have to be designed to be more powerful. A first attempt to exploit the result is the transmission strategy that we (together with Béla Bollobás and his collaborators) proposed in [6].

2.3 Energy and delay balancing

In most applications of networked embedded systems, traffic is routed to a common base station or access point. This causes *traffic accumulation* towards the base station, which is a serious problem affecting the lifetime of the network. In [7] we studied this problem and suggested different solutions.

2.4 Long-hop vs. short-hop routing

It is a non-trivial problem to determine the optimum hop length for routing in large sensor networks. In [2, 8, 9] we compared routing schemes with different hop lengths and concluded that often, *long-hop* routing offers superior performance.

2.5 Queueing analysis

For applications requiring QoS guarantees, a detailed *queueing analysis* is needed for performance analysis. While prior work was mostly restricted to delay means, we derived the end-to-end delay *distributions* [10–12] for certain network topologies.

2.6 Power amplifier characteristics

Often, it is wrongfully assumed that a reduction of the RF transmit power reads to a proportional reduction in the actual energy consumption. Since this is not true for practical power amplifiers, we carried out a detailed analysis in [13] that showed that transmission schemes without power control often offer better performance.

2.7 Managing uncertainty

To cope with the uncertainties in large systems of networked embedded sensors, we proposed the following approaches: Redundancy, Adaptivity, Diversity, Opportunism (RADO). We investigated the benefits of opportunistic channel access in [14] and proposed a simple formalism for the design and analysis of time and path diversity schemes in [15].

[2] With an average out-degree of about 1, an arbitrary node cannot reach anybody with probability $e^{-1} \approx 36\%$.

References

[1] M. Haenggi, “Efficient Routing in Wireless Networks with Random Node Distribution,” in *IEEE International Symposium on Information Theory (ISIT’04)*, (Chicago, IL), p. 17, June 2004. Available at `http://www.nd.edu/~mhaenggi/pubs/isit04.pdf`.

[2] M. Haenggi, “On Routing in Random Rayleigh Fading Networks,” *IEEE Transactions on Wireless Communications*, 2004. Accepted for publication. Available at `http://www.nd.edu/~mhaenggi/pubs/routing.pdf`.

[3] X. Liu and M. Haenggi, “Throughput Analysis of Fading Sensor Networks with Regular and Random Topologies,” *EURASIP Journal on Wireless Communications and Networking*, 2004. Submitted to Special Issue on Wireless Sensor Networks.

[4] X. Liu and M. Haenggi, “On Random and Quasi-Regular Sensor Networks,” *ACM Transactions on Sensor Networks*, 2005. Submitted.

[5] M. Haenggi, “On the Local Throughput of Large Interference-Limited Wireless Networks,” in *39th Annual Conference on Information Sciences and Systems (CISS’05)*, 2005. Submitted.

[6] P. Balister, B. Bollobás, M. Haenggi, and M. Walters, “Fast Transmission in Ad Hoc Networks,” in *IEEE International Symposium on Information Theory (ISIT’04)*, (Chicago, IL), p. 19, June 2004.

[7] M. Haenggi, “Energy-Balancing Strategies for Wireless Sensor Networks,” *IEEE Transactions on Circuits and Systems I*, Nov. 2004. Submitted for publication. Available at `http://www.nd.edu/~mhaenggi/pubs/tcas05.pdf`.

[8] M. Haenggi, “Twelve Reasons not to Route over Many Short Hops,” in *IEEE Vehicular Technology Conference (VTC’04 Fall)*, (Los Angeles, CA), Sept. 2004. Available at `http://www.nd.edu/~mhaenggi/pubs/vtc04.pdf`.

[9] M. Haenggi, “Routing in Ad Hoc Networks: A Case for Long Hops,” *IEEE Communications Magazine*, 2004. Submitted to Series on Ad Hoc and Sensor Networks.

[10] M. Xie and M. Haenggi, “Performance Analysis of a Priority Queueing System Over Rayleigh Fading Channels,” in *41st Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, IL), Oct. 2003. Available at `http://www.nd.edu/~mhaenggi/pubs/allerton03.pdf`.

[11] M. Xie and M. Haenggi, “Delay Analysis of Wireless Sensor Networks for Different MAC Schemes,” in *The Fourth International Conference on Information Processing in Sensor Networks (IPSN’05)*, (Los Angeles, CA), Apr. 2005. Submitted.

[12] M. Xie and M. Haenggi, “Study of QoS-Guaranteed Wireless Line Networks with TDMA and Slotted ALOHA,” in *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc’05)*, 2005. Submitted.

[13] M. Haenggi, “The Impact of Power Amplifier Characteristics on Routing in Random Wireless Networks,” in *IEEE Global Communications Conference (GLOBECOM’03)*, (San Francisco, CA), Dec. 2003. Available at `http://www.nd.edu/~mhaenggi/pubs/globecom03.pdf`.

[14] J. Venkataraman and M. Haenggi, “Optimizing the Throughput in Random Wireless Ad Hoc Networks,” in *42st Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, IL), Oct. 2004.

[15] M. Haenggi, “Analysis and Design of Diversity Schemes for Ad Hoc Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, Jan. 2005. To appear. Available at `http://www.nd.edu/~mhaenggi/pubs/jsac_adhoc.pdf`.

APPENDICES A, B, C, D

APPENDIX A
AUGUST 20, 2003 FIELD EXPERIMENT REPORT TO DARPA

Red Force Tagging Status Report

Objective: The objective of the Red Force Tagging Field Experiment was to recover data from a mobile sensor that was emplaced upon an individual and then use that data to reconstruct the individual's trajectory. Data recovery was accomplished using a network of MICA2 motes running a set of NEST middleware services that supported large data transfers. In order to achieve this objective, the Notre Dame team developed the mobile sensor (called the TagMote), developed the middleware services supporting data transfers from the TagMote to network motes and then to a base station (Relay), developed the trajectory reconstruction algorithms, and integrated all of these components into a single working system.

Description: The experiment was conducted on August 20, 2003 at MacDill Air Force Base (AFB). The experiment had a randomly chosen subject from the audience wear the TagMote and had the subject take a three minute walk in the vicinity of Lewis Lake (MacDill AFB). The subject then returned to the NEST mote field and the TagMote downloading its data into a network consisting of 16 MICA2 motes. The downloaded data was stored to the MICA2's flash memory. During the download a base station listened to the network traffic, picked out the downloaded data packets from the TagMote, and stored this data to a file. After the TagMote's data was downloaded, that data was then *exfiltrated* from the network motes to a Relay connected to a PC. The exfiltrated data packets were then written to a file. After both data sets were recovered, the subject's trajectory was reconstructed from the downloaded packets and the exfiltrated packets.

Performance: The experiment was run four times for two separate groups of visitors. The principal performance measures were:

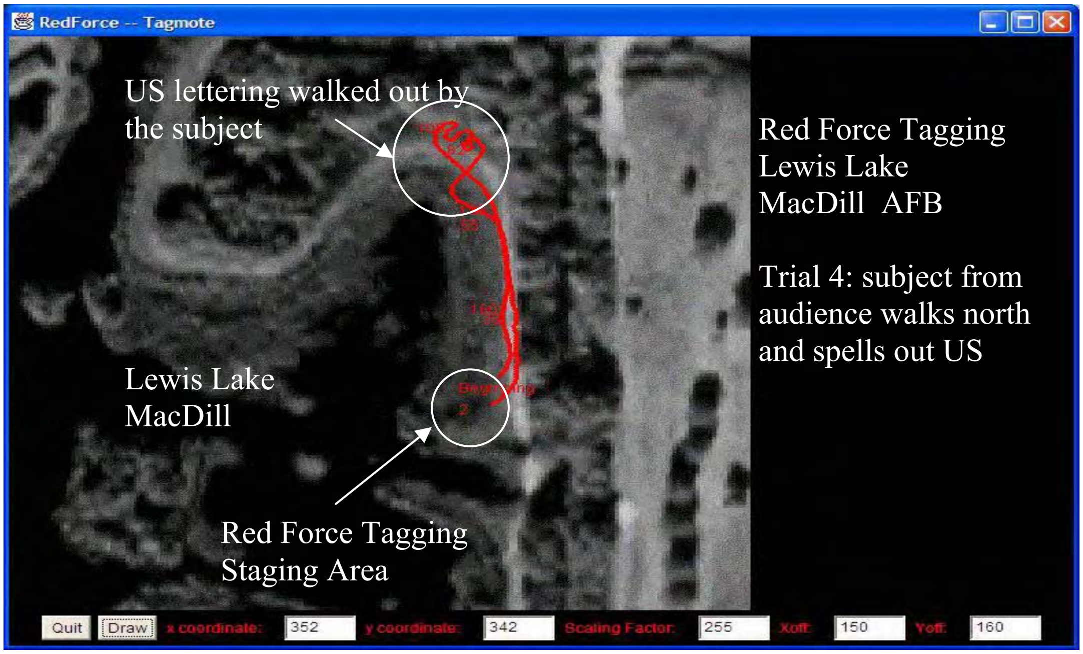
1. Exfiltration Performance Measure (Recovery Rate):
   The number of exfiltrated data lines / number of downloaded data lines
2. Download Performance Measure (Time, Data Rate):
   Download Time, number of downloaded data bytes / second
3. Exfiltration Performance Measure (Time, Data Rate):
   Exfiltration Time, number of exfiltrated data bytes / second.
4. Trajectory Reconstruction Accuracy:
   Plot of reconstructed trajectories viewed against aerial map of region.

The table shows the observed performance metrics for the 4 trials conducted during the experiment. The table shows that we were successful in recovering 99% of the TagMote's Trajectory data, with download times on the order of 3 minutes / 3 minute walk (70 data bytes / second) and exfiltration times on the order of 1 minute / 3 minute walk (200 data bytes / second).

	Recovery rate exfiled/download	Download Time (sec)	Download Throughput (bytes/sec)	Exfiltration Time (seconds)	Exfiltration Throughput (bytes/sec)
Trial 1	99.9%	200	84	69	242
Trial 2	103.7%	189	72	111	127
Trial 3	100.0%	189	79	67	222
Trial 4	99.9%	184	63	55	213
Average		190.5	74.5	75.5	201

Trial 2 exhibited some interesting things. First the recovery rate (ratio of exfiltrated versus downloaded packets) was greater than 100 percent. This is because the base station listening to the download was unable to hear all of the downloaded packets. This discrepancy between downloaded and exfiltrated data is shown in the reconstructed trajectories in the appendix. The reconstruction from the exfiled data in trial 2 exhibits a small loop at the upper end of the walk. This loop is not present in the reconstruction made from the download packets overheard by our base station. Clearly the packets associated with this loop are the packets that our base station did not overhear. The other interesting thing to note is that the exfiltration time for trial 2 was longer than for the other runs. This is because the majority of the data in trial 2 was exfiltrated over a two-hop, rather than a one-hop connection.

The reconstructed trajectories had an accuracy that was consistent with earlier predictions of 20 meters over 1 kilometer of distance traveled. The following picture shows the aerial view of the last trial's trajectory reconstructed from the exfiltrated data. In this particular field run, the subject walked north along the road bordering Lewis Lake and then spelled out US during his walk. As can be seen from the attached figure, the reconstructed trajectories clearly show this feature in the subject's walk. On the basis of these results, we felt the field experiment successfully met its objective. Pictures of the reconstructed trajectories for the other 3 trials are attached to the end of this report.



US lettering walked out by the subject

Red Force Tagging
Lewis Lake
MacDill AFB

Trial 4: subject from audience walks north and spells out US

Lewis Lake
MacDill

Red Force Tagging
Staging Area

x coordinate: 352 y coordinate: 342 Scaling Factor: 255 Xoff: 150 Yoff: 160

Caveats: In order to facilitate a smoothly flowing demonstration, several simplifications were adopted in the experiment. These simplifications and the justification for using them are itemized below.

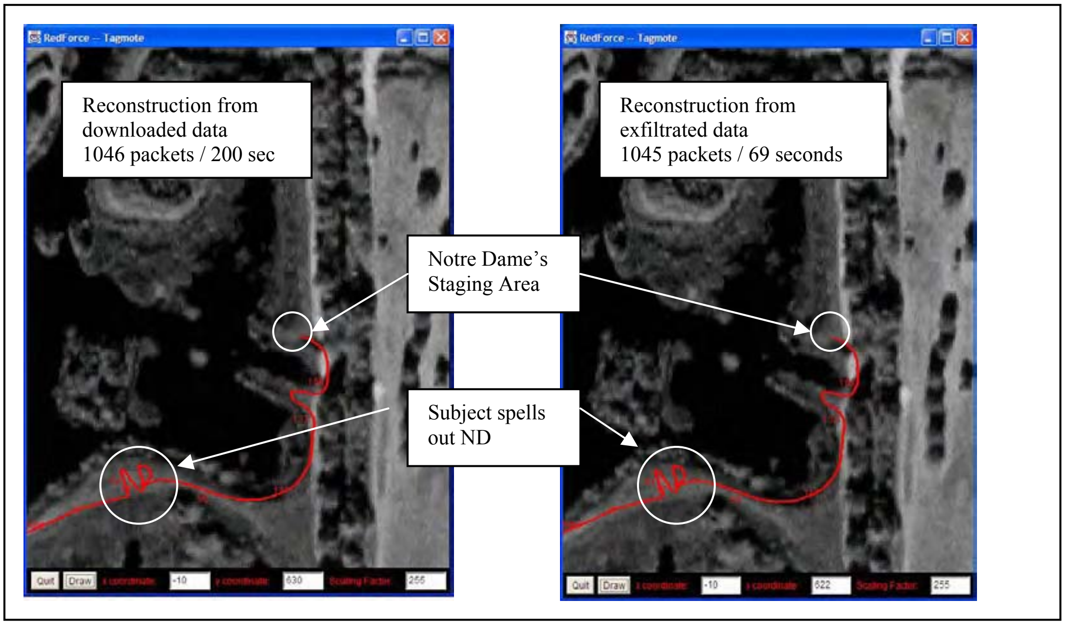
1. The subject's walk was restricted to 3 minutes. This was done so the audience didn't need to wait for a long period of time before seeing the reconstruction. The actual TagMote is capable of storing 2 hours of walking data.
2. The subject was requested to stand still during the downloading process. This was also done to keep the length of the experiment reasonable. While we have done numerous experiments with the subject moving through the field during a download, we have usually found that this can dramatically lengthen the download time. The precise increase in duration is difficult to predict and depends greatly upon the subject's movements and environmental conditions.
3. Network size was restricted to 16-25 motes. This was done to reduce the length of the exfiltration time. For larger networks, we would expect a preponderance of multi-hop routes to form. As mentioned above, the exfiltration rate for a single hop is 200 data bytes/second. This time increases linearly with the number of hops in the route from the NestMote to the Relay. So we restricted our network size to ensure a preponderance of single and two-hop routes. This simplification greatly reduced the length of time the audience had to wait for the exfiltrated data.
4. The experiment did not use a localization service. Localization in the NEST middleware is a difficult problem and we simply assumed that it was already in place. In practice, we would also want to implement some sort of localization service.
5. Trajectory Reconstruction was not completely automated. We viewed this as a "feature", rather than a simplification. Trajectory reconstruction quality is governed by a number of algorithm parameters that are dependent on the subject and the site location. In practice, it is usually impossible to ensure that all of these parameters are known before hand. We therefore designed our trajectory reconstruction tool so a user could interactively adjust the reconstruction parameters to force the trajectory to match known waypoints that the subject passed.

Many of the preceding simplifications were adopted to enhance the flow of the field experiment for the audience. We felt that none of these simplifications fundamentally restricted the experiment's scope.

Reconstructed Trajectories

The following pages show the reconstructed trajectories from the 4 field experiments.

Experiment 1 (walk from UVA to ND's site)

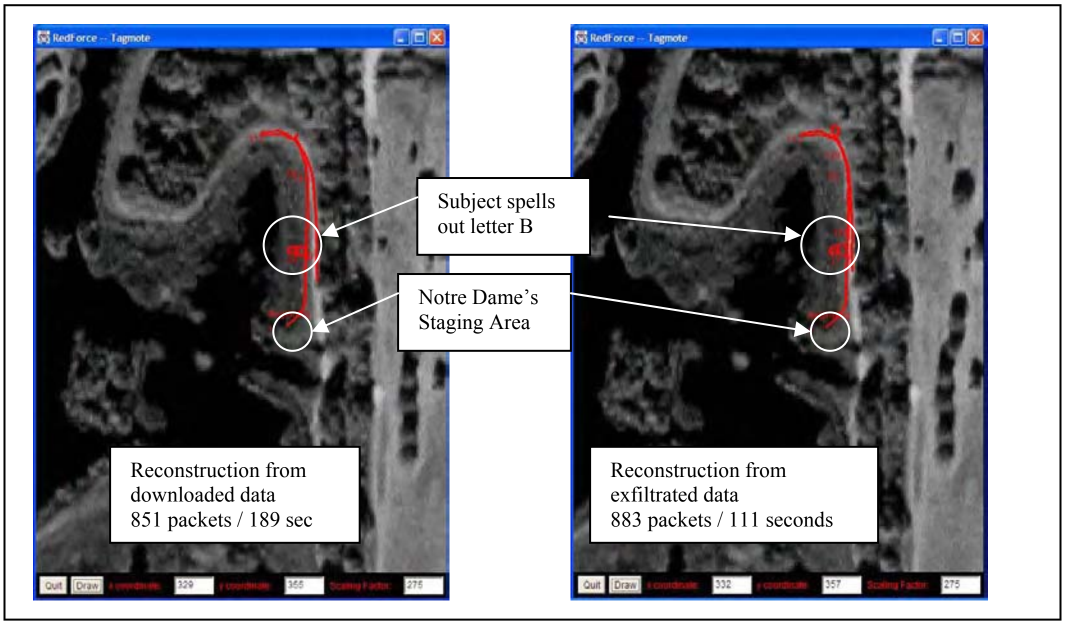

Reconstruction from downloaded data 1046 packets / 200 sec

Reconstruction from exfiltrated data 1045 packets / 69 seconds

Notre Dame’s Staging Area

Subject spells out ND

Experiment 2 (subject’s walk north of ND’s site)

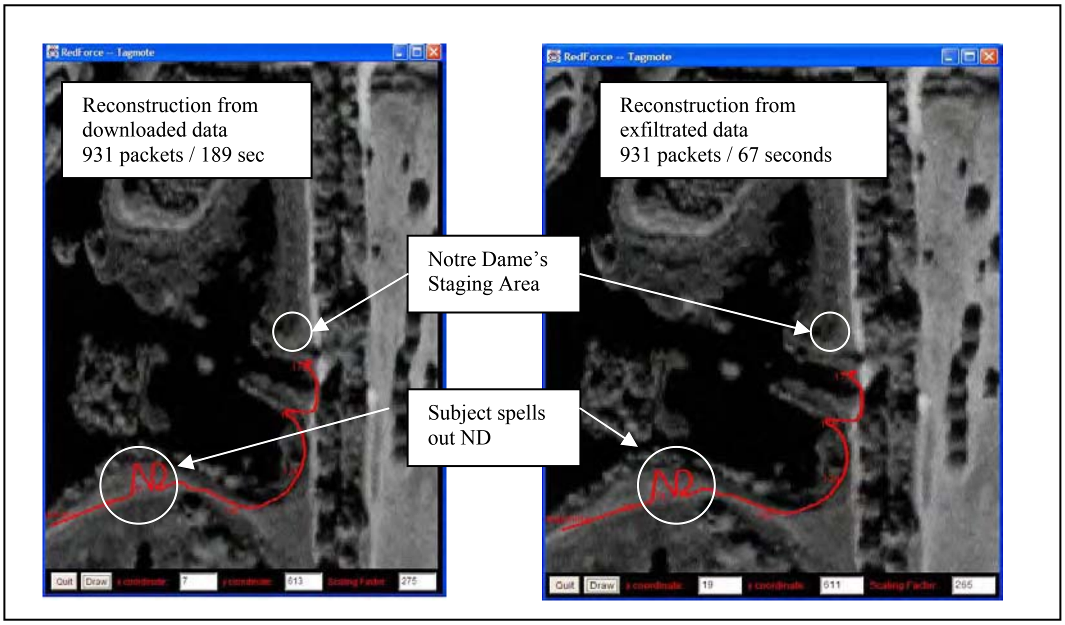

Subject spells out letter B

Notre Dame’s Staging Area

Reconstruction from downloaded data 851 packets / 189 sec

Reconstruction from exfiltrated data 883 packets / 111 seconds

Experiment 3 (walk from UVA’s site to ND’s site)



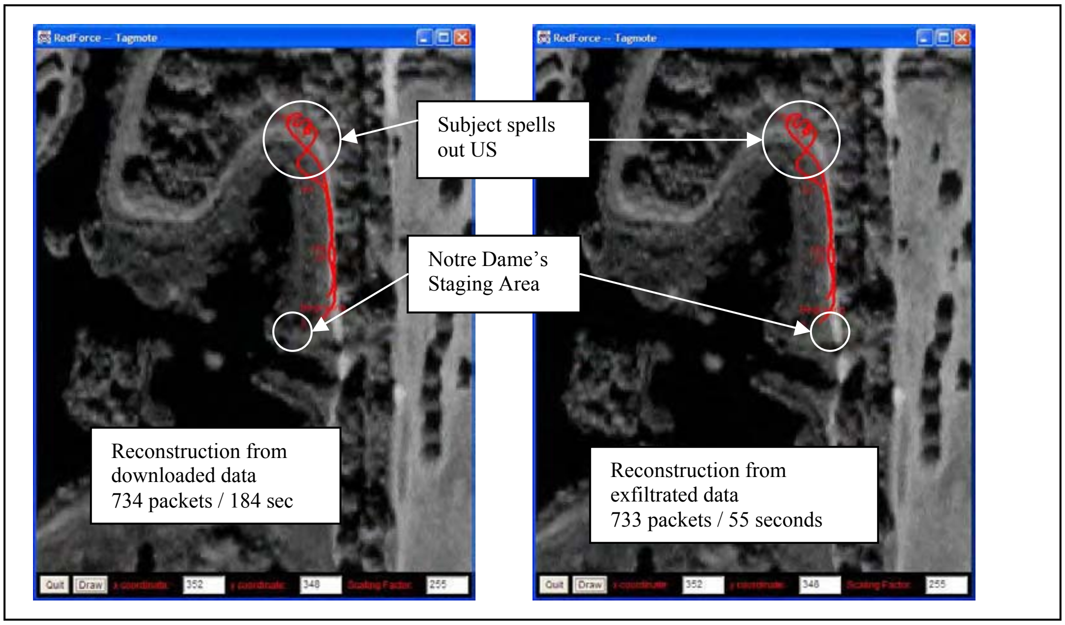
Reconstruction from downloaded data 931 packets / 189 sec

Reconstruction from exfiltrated data 931 packets / 67 seconds

Notre Dame's Staging Area

Subject spells out ND

Experiment 4 (walk north of ND's site)



Subject spells out US

Notre Dame's Staging Area

Reconstruction from downloaded data 734 packets / 184 sec

Reconstruction from exfiltrated data 733 packets / 55 seconds

APPENDIX B
Experimental Results (June 2004)

The Tagmote code has several working versions. The older versions moved gradually from a centralized (one processor) implementation that was simply sending raw data to the network, towards a distributed (two processor) implementation, in which data undergoes some real-time processing, including a data compression process. For a detailed discussion, see Sections 4 & 5.

Tagmote Design Innovations	**Download Time (% of walking time*)**	**Exfiltration Time (One-hop Case)**
One Processor (Aug.03 Demo)	106%	31%
Preproc. on master after flash storage	156%	31%
Preproc. on slave after flash storage	89%	31%
Preproc. on slave before flash storage	**44%**	**31%**
As before with differential compression	**28%**	**17%**
*: The total walking time with 2M bytes flash is up to 10 hours. In fact, the sustainable walking time is limited by the battery capacity.		

Table B.1 Experimental Results Comparison among Different Tagmote Versions

Appendix C: Publications

Lei Fang, Panos J. Antsaklis, Luis Montestruque, Brett McMickell, Michael Lemmon, Yashan Sun, Hui Fang, Ioannis Koutroulis, Martin Haenggi, Min Xie, and Xiaojuan Xie, "A Wireless Dead Reckoning Pedestrian Navigation System," ACM Workshop on Applications of Mobile Embedded Systems, Boston, MA, June 6, 2004. .

M.D. Lemmon, Q. Ling, and Y. Sun , “Overload Management in Sensor-Actuator Networks used for Spatially-Distributed Control Systems,” 1st ACM Conference on Embedded Network Sensor Systems (SenSys03), UCLA, Nov. 2003.
ABSTRACT
Overload management policies avoid network congestion by actively dropping packets. This paper studies the effect that such data dropouts have on the performance of spatially distributed control systems. We formally relate the spatially-distributed system's performance (as measured by the average output signal power) to the data dropout rate. This relationship is used to pose an optimization problem whose solution is a Markov chain characterizing a dropout process that maximizes control system performance subject to a specified lower bound on the dropout rate. We then use this Markov chain to formulate an overload management policy that enables nodes to enforce the "optimal" dropout process identified in our optimization problem. Simulation experiments are used to verify the paper's claims.

M. Haenggi, “On Routing in Random Rayleigh Fading Networks,” IEEE Transactions on Wireless Communications, 2004. Accepted for publication. Available at http://www.nd.edu/~mhaenggi/pubs/routing.pdf.

M. Haenggi, “Energy-Balancing Strategies for Wireless Sensor Networks,” IEEE Transactions on Circuits and Systems I, Nov. 2004. Submitted for publication. Available at http://www.nd.edu/~mhaenggi/pubs/tcas05.pdf.

M. Haenggi, “Twelve Reasons not to Route over Many Short Hops,” in IEEE Vehicular Technology Conference (VTC'04 Fall), (Los Angeles, CA), Sept. 2004. Available at http://www.nd.edu/~mhaenggi/pubs/vtc04.pdf.

M. Haenggi, “Analysis and Design of Diversity Schemes for Ad Hoc Wireless Networks,” IEEE Journal on Selected Areas in Communications, vol. 23, Jan. 2005. To appear. Available at http://www.nd.edu/~mhaenggi/pubs/jsac_adhoc.pdf.

A Wireless Dead Reckoning Pedestrian Tracking System

Lei Fang, Panos J. Antsaklis, Luis Montestruque, Brett McMickell, Michael Lemmon,
Yashan Sun, Hui Fang, Ioannis Koutroulis, Martin Haenggi, Min Xie, and Xiaojuan Xie
Department of Electrical Engineering
Univ. of Notre Dame, Notre Dame, IN 46556, USA
{lfang,antsaklis.1}@nd.edu

1. INTRODUCTION

The ability to track the position of the user is an essential part of many applications [7, 4, 2]. It is well known that Global Positioning System (GPS) is limited as a navigation aid by its inability to provide static heading and its lack of availability when used around obstructions (terrain or man-made) or in the presence of jamming. Therefore, it is necessary to develop a positioning system which can complement GPS in GPS-compromised areas, which is not a trivial task.

With advances in computation, communication and sensing capabilities, large scale sensor-based distributed environments are emerging as a predominant mobile computing infrastructure [1]. Hundreds or thousands of small, inexpensive and low-power sensors, such as Berkeley Motes [5], can be quickly deployed to monitor a vast field. In this work we make use of this mobile infrastructure and new product innovations to build a pedestrian tracking system (PTS) that is capable of working either indoors or outdoors. Fig. 1 illustrates the system architecture. The system involves a Dead Reckoning (DR) module, self-organizing wireless networks (sensor mote network) and a map database. Dead reckoning is used by DR modules to determine their positions and trajectories, and the sensor mote network is used to collect trajectory data and make them available at a base station for further processing. The DR module consists of the Leica Geosystems DMC-SX three axes accelerometer and magnetic compass (Leica Vectronix AG [6]) combined with a generic wireless controller board, with the radio, the processing and the power storage all integrated. The controller board for DR module and network nodes both use the open-source Berkeley Motes with the *TinyOS*. In the following, we call the DR module unit with the generic board, *NavMote*, and a network node, *NetMote*. The NetMotes play a critical role in NavMote calibration, trajectory data collection and exfiltration, all via wireless communication links. The trajectory data collected by NetMotes can be further exfiltrated to an information center for displaying, map matching, and other purposes.

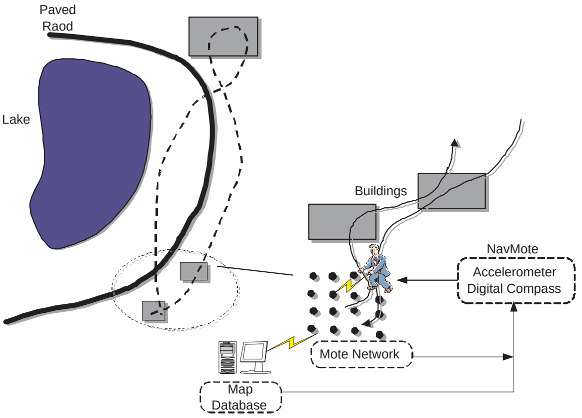


Figure 1: The pedestrian tracking system architecture

There are two challenges for the PTS: On the NavMote side, the resource-poor mote prevents using any computationally expensive algorithms, while size, weight and cost constraints need to be met. On the network side, a large amount of data need to be downloaded from the NavMote to one or more of (unreliable) NetMotes and forwarded to RelayMote with error-prone wireless channels. In this paper, we focus on the design challenges of the NetMote network. (For additional information regarding hardware, sensors, and DR algorithm, see [3].)

The present application differentiates itself from the traditional sensor network applications in several ways. First, large and sudden bursts of data must be delivered to the base station with a low latency. The transport of event impulses is likely to lead to varying degree of channel contention and network congestion. Second, these trajectory data are not redundant, requiring a near zero loss-rate. Third, but not last, the wireless links are highly unpredictable and unreliable, and bidirectional connectivity cannot generally be achieved. Commonly observed in the experiments are isolated nodes/clusters, leading to extended latency and even network deadlocks. To address these problems, we implement an IEEE 802.11 DCF (RTS/CTS/ACK) scheme to avoid channel contention, while guaranteeing a low-loss rate delivery with aggregated message ACKs. In our self-healing adaptive routing scheme, every node maintains a set of next-hop candidates. The "best" candidate is selected based on throughput estimate. Various deadlock cases due to missed messages are identified and countermeasures are taken to

WAMES '04, June 6, 2004, Boston, Massachusetts, USA.
Copyright 2004 ACM 1-58113-000-0/00/0004 ...$5.00.

eliminates these cases.

2. THE SYSTEM DESCRIPTION

2.1 Overview

A typical scenario involves the reconstruction of the space/ time path a subject has taken from an initial to a final location. While the NavMote carried by the subject is out of range of the network, it uses the flash memory to store the accelerometer and compass data in real-time. Custom algorithms provide compensation for adverse sensor orientation and calibration, and for collection with possible coding of data. When the subject enters again into radio range of the network, the data captured by NavMote are extracted automatically via wireless communication and deposited in the NetMotes's EEPROM. These data that are distributed within the network are then via multi-hop transmissions sent to a designated special "relay" or "destination" mote (*RelayMote*) for further processing. At this stage, we assume the initial location is known and localization service is available to provide NetMote positions.

In what follows, we discuss the technical consideration that underpin the design of the NetMote network while the detailed design is presented in the next subsection.

The media access control plays a significant role in the performance of managing bursts of data in a wireless shared medium. A number of sensor networks use CSMA or variants for medium access. For example, the TinyOS platform uses a simple CSMA MAC. CSMA suffers from the well-known hidden terminal problem in multihop environment. IEEE 802.11 utilizes an RTS/CTS exchange to eliminate hidden terminals. Although the signaling cost of the RTS/CTS exchange is high for sensor networks where the packet size is small, the scheme is justified for the PTS application due to the burst traffic characteristics; for a three-minute walk, NavMote has approximately 12 KBytes (compressed) data, which are often downloaded to several NetMotes. Every NetMote then tries to transmit the data in its buffer of size 128×16 Bytes in one RTS/CTS exchange. Hence the number of RTS/CTS exchanges is quite small compared to the data size. Random delays are also introduced in addition to backoff as suggested in [9]. To further reduce energy cost, ACK is used in an aggregated manner, not for every single packet.

In the course of network operation, NetMote and link failures may lead to arbitrary and unsupport topology. The RelayMote is required to use the active topology for extraction of information from the network and distribution of control packets to the NetMotes. The algorithm which we have used to satisfy the requirements of reliable routing is a distributed leader election algorithm. The algorithm is used to construct a minimum-hop spanning tree, such that each NetMote knows a "parent node" closer to the RelayMote. At each hop, the NetMote receives a packet and retransmits it to upper level. As a preventive measure, a periodic recomputation of minimum spanning is triggered by the RelayMote to account for any changes in the topology. Furthermore, all the NetMotes keep the number of transmission retries as the throughput estimate. The best parent node, which has least number of retries, is selected. Max retried times is also set to prevent from forwarding data to inactive or dead NetMotes. If there is no "good" parent available, then the NetMote holds data till the next spanning tree update. The watchdog function is another essential part of the scheme to ensure a robust routing protocol. For instance, silent motes caused by CTS can be unlocked automatically even in case of missing messages or software task deadlines.

2.2 NetMote Network Middleware Services

The software supporting the PTS application is written using TinyOS/NesC [8]. The software is organized into a stack consisting of three layers; the application, middleware and operating system or O/S layers. There are three primary *application services*: *Coordinator*: This service manages the interaction of the other application services; *Dump*: This service downloads trajectory data from the NavMote to the NetMote's EEPROM; *Exfile*: This service streams data in a NetMote's EEPROM to the network's RelayMote (base station).

There are five primary *middleware services*: *Ping*: NavMote uses this service to determine if it is in the vicinity of a NetMote; *Backbone*: This service builds and maintains a robust minimum hop spanning tree from all NetMotes to the RelayMote; *Clock Synchronization*: This service maintains a global clock variable across all nodes in the network; *Localization*: This service initializes and maintains a variable representing the NetMote's physical position; *Telemetry*: This service periodically sends packets down the network's backbone to the RelayMote. The telemetry packets contain information about the network's current configuration. This service is used by the RelayMote to build a picture of the entire network.

The *O/S services* are software components interfacing directly to the mote's physical resources such as the UART, radio, sensors, clock, random number generator, and EEPROM (logger).

The services in the middleware layer are responsible for setting up and maintaining the network infrastructure required to recover data from the NavMote in a flexible and reliable fashion. As soon as a NetMote is reset, it starts up these middleware services in a specific order. The *Clock Synchronization* and *Localization* services are started first in order to initialize the mote's clock and location variables. Once these services have stabilized, a signal is issued which starts the *Backbone* service. This service automatically builds a minimum hop spanning tree from all NetMotes to the RelayMote. The service is designed to detect changes in link quality that would adversely effect the network connectivity. Upon detecting such changes, the backbone service reconfigures its spanning tree to restore network connectivity. Finally, once the backbone service has stabilized, it issues a signal which starts the telemetry service. The telemetry service simply sends packets to the RelayMote that provide the user with a global view of the network's connectivity.

The NavMote uses the *Ping* service to find the network. Upon finding the network, it uses the application layer's *Dump* service to download the trajectory data to the network. The dump service works in an opportunistic manner, dumping as much data as possible to the nearest available NetMote's EEPROM. The download then switches to another download should the original dump stream be interrupted. Having multiple NetMotes store the NavMotes data is desirable since a single NetMote may not have enough EEPROM to store all of the trajectory data and the subject may be moving from being close to one NetMote to being closer to a different NetMote.

The *Exfile* service is used to transmit data logged in the NetMote's EEPROM to the RelayMote. The *coordinator* on the NetMote ensures that the *Dump* and *Exfile* service operate in a mutually exclusive manner. In this way, the Exfile service does not begin until the download from the NavMote has stopped. Upon starting, the Exfile service begins reading data stored in EEPROM and storing it in a 128 line (16 Bytes/line) buffer. Once this buffer is full, the service issues a request-to-send (RTS) to the next-hop on the backbone's spanning tree. All nodes (except the sender and next-hop) hearing the RTS are immediately set to a dormant state, thereby establishing a *basic service cell* (BSC) in which the sender and next-hop can communicate with little interference from other NetMotes. Upon receiving a clear-to-send (CTS) from the next-hop, the Exfile service begins transmitting the buffered data, line by line, to the next-hop. Every four transmitted lines is acknowledged (ACK) by the next-hop. Should the sender miss an ACK or the next-hop miss an expected message, then they issue a clear (CLR) message that resets the basic service cell so it is ready for the next RTS message.

The Exfile service essentially streams data from all NetMotes with logged data to the RelayMote. The trajectory data transmitted by the Exfile service follows the same spanning tree used by the *Telemetry* service packets. The final destination for both message streams is the RelayMote. Upon being caught by the RelayMote, the packets are forwarded over the RelayMote's UART to a PC that is running a Java graphical user interface (GUI) called *NetConsole*. The GUI displays telemetry packets that are forwarded to the RelayMote from the network. The data contained in these packets allow the GUI to display the NetMote and NavMote position and neighborhoods. The data also allow the GUI to display the routes and connectivity between NetMotes. Finally, the NetConsole dumps the decoded data packets to a file for subsequent post-processing.

3. CONCLUSIONS

In this paper, we describe the development of a pedestrian tracking system that uses three-axes accelerometer and magnetic compass (Leica Geosystems) and a mote network consisting of Berkeley motes.

The system was examined in both indoor and outdoor environments. One group of experiments took place in swamp terrain. The subjects took 5-minute or 3-minute walks through the test course. The principal performance measures are:

1. Exfiltration Performance Measure (Rec Rate): The number of exfiltrated data lines / number of downloaded data lines.
2. Download Performance Measure (DTime, DTput): Download Time, number of downloaded data bytes/sec.
3. Exfiltration Performance Measure (ETime, ETput): Exfiltration Time, number of exfiltrated data bytes/sec.
4. Trajectory Reconstruction Accuracy: Plot of reconstructed trajectories viewed against aerial map of region.

Table 1 shows the observed performance metrics for the 4 trials conducted during the test. A total of 21 NetMotes are deployed in a 3 × 7 grid, with a grid size around 7 feet. The table shows that we were successful in recovering 99% of the NavMote's trajectory data, with download times on the order of 3min/3min walk (70 data bytes/sec) and exfiltration times on the order of 1min/3min walk (200 data bytes/sec). Extensive experiments show that the accuracy of walk distance estimate is with ±3% and heading accuracy ±1°.

	Rec Rate	DTime (sec)	DTput (b/s)	ETime (sec)	ETput (b/s)
1	99.9%	200	84	69	242
2	100.0%	189	72	111	127
3	100.0%	189	79	67	222
4	99.9%	184	63	55	213
Avg.		190.5	74.5	75.5	201

Table 1: System Performance

Our experiences tell us that sensor nodes and wireless links are much unpredictable and unreliable than one might think. Hence a simple yet robust MAC scheme/routing protocol are critical to any sensor network application. We focus more on the high throughput and low loss-rate aspects, without fully addressing the energy-aware network operation, which is in some sense more important for the sensor network applications. Other practical challenges include node localization, network security, privacy, and authentication. We believe however that our development offers a vision of future mobile environments that may emerge once ubiquitous wireless coverage becomes available.

Acknowledgment

The support of the DARPA/IXO-NEST Program (AF-F30602-01-2-0526) and of the National Science Foundation (CCR-02-08537 and ECS-02-25265) is gratefully acknowledged.

4. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Networks*, vol. 38, pp. 393-422, 2002.

[2] J. Elwell, "Inertial navigation for the urban warrior," in *Proc. SPIE*, vol. 3709, 1999, pp. 196-204.

[3] L. Fang, P. J. Antsaklis, et. al, "Design of a wireless dead reckoning pedestrian navigation system - The NavMote experience," Technical Report, ISIS Lab, University of Notre Dame. [Online]. http://www.nd.edu/~isis/techreports/isis-2004-001.pdf

[4] H. Hashimoto, K. Magatani, and K. Yanashima, "The development of the navigation system for visually impaired persons," in *Proc. 23rd Annual EMBS Int. Conf.*, 2001, Istanbul, Turkey, pp. 1481-1483.

[5] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12-24, 2002.

[6] "DMC-SX Digital Magnetic Compass - Operator Manual," Leica Vectronix AG, Switzerland.

[7] T. E. Starner, *Wearable Computing and Contextual Awareness*, Ph.D. Thesis, MIT, 1999.

[8] TinyOS - A component-based OS for networked sensor regime. [Online]. http://webs.cs.berkeley.edu/tos/

[9] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," *MOBICOM 2001*, Rome, Italy, pp. 221-235.

OVERLOAD MANAGEMENT IN SENSOR-ACTUATOR NETWORKS USED FOR SPATIALLY-DISTRIBUTED CONTROL SYSTEMS

Michael Lemmon [*]
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
lemmon@nd.edu

Qiang Ling
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
qling@nd.edu

Yashan Sun
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
Yashan.Sun.21@nd.edu

ABSTRACT

Overload management policies avoid network congestion by actively dropping packets. This paper studies the effect that such data dropouts have on the performance of spatially distributed control systems. We formally relate the spatially-distributed system's performance (as measured by the average output signal power) to the data dropout rate. This relationship is used to pose an optimization problem whose solution is a Markov chain characterizing a dropout process that maximizes control system performance subject to a specified lower bound on the dropout rate. We then use this Markov chain to formulate an overload management policy that enables nodes to enforce the "optimal" dropout process identified in our optimization problem. Simulation experiments are used to verify the paper's claims.

Categories and Subject Descriptors

C.3 [**Special Purpose and Application-Based Systems**]: Real-time and Embedded Systems

General Terms

Theory

Keywords

Networked Control Systems, Spatially-Distributed Systems, Overload Management, Sensor-Actuator Networks

1. INTRODUCTION

A *sensor-actuator network* is a collection of sensors and actuators that exchange information over a shared communication network. Such networks can be used in active structural acoustic control [3] or other spatially distributed control systems [1] [2]. An example of such a system is shown in Figure 1. This system [1] is a linear array of masses that are linked together by springs. Each mass (node) has an embedded processor attached to it that can measure the node's local state (position) and then transmit that information to its neighboring nodes. The n_2th node's position at time n_1 is denoted as x_{n_1,n_2}. This transmitted information is then used by the embedded processor to compute a control input. The control input to the n_2th node at time n_1 is denoted as u_{n_1,n_2}. Distributed controllers for such systems can be designed using a host of methods. Most of these controllers, however, assume there is no loss of data through the network. This paper examines the effect dropouts have on overall control system performance and devises policies for optimally managing these dropouts.

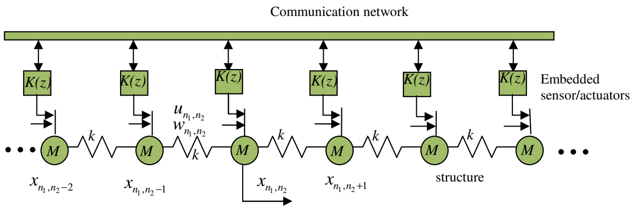


Figure 1: Distributed Control System and Sensor-Actuator Network

Data dropouts occur for many reasons, but this paper confines its attention to networks in which 1) data are dropped due to link failures or 2) data are purposefully dropped in order to stay within a specified throughput allocation where throughput is defined to be the average number of feedback measurements per second. The first source of data dropouts cannot be controlled by the nodes and we assume that such dropouts occur at a rate f that is statistically independent of the data source. This model for link failure may be appropriate for networks in which small sets of nodes share the same physical medium. Examples of such networks include RF (radio) wireless networks as well as wired networks in

[*] We gratefully acknowledge the partial financial support of the Army Research Office (DAAG19-01-0743), the National Science Foundation (NSF-CCR02-8537, NSF-ECS02-25265) and DARPA's NEST program (F30602-01-2-0526)

SenSys'03, November 5–7, 2003, Los Angeles, California, USA.

which a node is connected by wire to it nearest neighbors. In such networks, the most likely cause for link failure arises from collisions between packets in the medium.

The second source of dropouts occurs because overload management policies require nodes to purposefully drop packets to prevent network congestion. In particular, the network allocates a portion of its available throughput to each node. Each node then selectively drops packets to stay within this throughput allocation. Previous work in overload management [9] has focused on the use of heuristic policies such as the (m, k)-firm guarantee rule. This paper provides an alternative approach to overload management that has provable guarantees on the attainable level of application performance.

In this paper, "application performance" refers to the performance delivered by a spatially distributed control system. Figure 2 illustrates a control system that consists of a controller $K(z)$ and plant $G(z)$ that are connected in a feedback loop. In our case, the feedback loop is implemented over a communication network. The input to the control system is a signal, w, that represents either an exogeneous noise or disturbance that is exciting the plant. The controller $K(z)$ minimizes the impact that this disturbance w has on the system's output, y. In particular, we seek to minimize the *power* in this output signal, y. We sometimes refer to this as a *generalized regulator problem.*

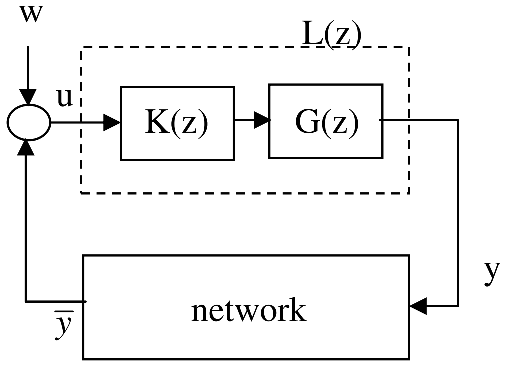


Figure 2: Generalized Regulator over a Network

The control system shown in Figure 2 is an abstract model of the physical system shown in Figure 1. For the physical system, the *plant* is the structure consisting of masses and springs. The *controller* is implemented in the embedded processors generating the input signals driving the individual masses. The *network* is the communication channel between the embedded processors. Essentially we can view this as an acoustic vibration suppression problem. The external signal w is a noise source that is exciting the physical structure. The embedded processors measure the resulting vibration in the structure and use these measurements to generate a control that tries to reduce that vibration. So the measure of this control system's performance is the average power in the physical structure's state signal.

This paper examines the impact that communication network Quality-of-Service (as measured by feedback measurement throughput) has on the control system's performance (as measured by the power in the structure's vibration). In particular, traditional control system design assumes that feedback measurements are being received by the controller at regular time intervals. Requiring that all feedback measurements are successfully transmitted over the network results in a hard real-time control system. Meeting hard real-time deadlines is extremely difficult in communication networks where several embedded processors share the same communication channel. This paper breaks the tyranny of periodic hard real-time scheduling by asking and answering the following questions. Can we characterize the control system's performance as a function of the average throughput rate? Is it possible to distribute dropped feedback measurements in a way that has the least impact on application (control system) performance? The answer to both questions is in the affirmative and this paper shows how those answers apply in sensor-actuator networks used to control the spatially distributed system shown in Figure 1.

The remainder of this paper is organized as follows. Section 2 presents the spatially distributed system studied in this paper. Section 3 discusses prior work relating control system performance and data dropouts. This prior work only pertains to single control loops, so section 4 extends that work to the spatially distributed system in figure 1. Section 5 uses these results to formulate an "optimal" overload management policy and the behavior of this policy is demonstrated on a simulation of a 27-node structure. Final remarks are found in section 6. The theorems' proofs will be found in section 7.

2. SPATIALLY DISTRIBUTED CONTROL SYSTEM

This paper confines its attention to the spatially distributed system shown in Figure 1. We first assume that the continuous-state dynamics of the system have been discretized in time. So we let x_{n_1,n_2} denote the state of the n_2th node at time instant n_1. The state x is a 2-vector characterizing the position and velocity of the node with respect to its equilibrium position. The discrete state satisfies the following recursive equations,

$$\begin{aligned} x_{n_1+1,n_2} &= Ax_{n_1,n_2} + B(x_{n_1,n_2-1} + x_{n_1,n_2+1}) \\ &\quad +F(u_{n_1,n_2} + w_{n_1,n_2}) \\ z_{n_1,n_2} &= Cx_{n_1,n_2} \end{aligned} \tag{1}$$

for $n_1 \geq 0$ and any n_2. z_{n_1,n_2} is an output signal that is used to characterize overall system performance. A, B, C and F are appropriately dimensioned real-valued matrices. There are two inputs to this equation; the disturbance w_{n_1,n_2} and the control u_{n_1,n_2}. The disturbance is a zero-mean white noise process in both time and space. The control input is computed by the embedded processor.

The examples in this paper confine their attention to 1-dimensional structures in which there is only a single spatial-variable, n_2. Note that this work may be easily extended to 2-d chains through the introduction of an additional spatial variable. Detailed models for such systems will be found [2]

Each node has a processor attached to it. The processor measures the node's local state x_{n_1,n_2} and it transmits this information to its neighbors upon request. We assume that the nodes are synchronized in time and that in each sampling interval the node decides whether or not to access its neighbor's state. This means that a node first "requests" that its neighbors send data to it and then the processor computes its control input u_{n_1,n_2} upon receiving this data. If neighboring state information has been received, then the control input is computed according to the following equation,

$$u_{n_1,n_2} = K_0 x_{n_1,n_2} + K_1(x_{n_1,n_2-1} + x_{n_1,n_2+1}) \qquad (2)$$

where K_0 and K_1 represent control gain matrices that have been chosen by the control engineer. Since our network may occasionally drop packets, the processor needs to use a different control signal if the neighboring state data is not received. In this case, the processor simply sets $u_{n_1,n_2} = 0$.

Data will always be dropped by the network for two reasons. The first reason is that the medium is unreliable. A transmitted packet has a finite probability f of being lost due to link failure. This probability is assumed to be statistically independent from the state of the packet's source. Dropouts will also occur because a node explicitly decides NOT to request neighboring state measurements. This occurs because an *overload management policy* requires nodes to drop a certain percentage of packets when the network is congested. In particular, the network allocates a specified amount of its throughput to each node which we characterized as a lower bound, ε_{d}, on the node's actual dropout rate. The size of ε_d depends on the amount of network congestion. Overload management through packet dropping clearly has an adverse impact on overall application performance. This is particularly true for hard real-time feedback control systems. This paper determines an overload management policy that is "optimal" in that it maximizes application (control system) performance while ensuring the dropout rate does not drop below ε_d.

Because dropouts cause us to switch between two different control laws, the system's state space model takes the form of a *jump linear system* [7]. In particular, let's define a *dropout process* that is denoted as d_{n_1,n_2}. It is a binary random process in which $d_{n_1,n_2} = 1$ if a dropout occurs and is zero otherwise. Under the dropout process, our system equations take the form,

$$\begin{aligned} x_{n_1+1,n_2} &= A_{n_1,n_2} x_{n_1,n_2} + B_{n_1,n_2}(x_{n_1,n_2-1} + \\ &\quad x_{n_1,n_2+1}) + F w_{n_1,n_2} \\ z_{n_1,n_2} &= C x_{n_1,n_2} \end{aligned} \qquad (3)$$

where A_{n_1,n_2} and B_{n_1,n_2} are matrix valued random processes such that

$$A_{n_1,n_2} = \begin{cases} A_0 = A + FK_0 & \text{if no dropouts occur} \\ & (i.e., d_{n_1,n_2} = 0) \\ A_1 = A & \text{if a dropout occurs} \\ & (i.e., d_{n_1,n_2} = 1) \end{cases}$$

$$B_{n_1,n_2} = \begin{cases} B_0 = B + FK_1 & \text{if no dropouts occur} \\ & (i.e., d_{n_1,n_2} = 0) \\ B_1 = B & \text{if a dropout occurs} \\ & (i.e., d_{n_1,n_2} = 1) \end{cases}$$

Application performance will be measured by the average power in the control system's output signal. This is a standard measure of performance for regulation problems. In our case, we want to suppress the effect that the disturbance w_{n_1,n_2} has on the system's shape. In particular we assume that w is a white noise process whose covariance matrix is

$$R = \mathbf{E}\{w_{n_1,n_2} w_{n_1,n_2}^T\}$$

The control objective is to minimize the noise power in the node's state. So a natural measure of application performance is the average power, $\|z\|_{\mathcal{P}}^2$, in the node's output signal z_{n_1,n_2}. This power is usually written as

$$\|z\|_{\mathcal{P}}^2 = \text{Trace}\mathbf{E}\{z_{n_1,n_2} z_{n_1,n_2}^T\} = \text{Trace}\left[C\overline{P}_0 C^T\right]$$

where $\overline{P}_0$ is the covariance matrix

$$\overline{P}_0 = \mathbf{E}\{x_{n_1,n_2} x_{n_1,n_2}^T\} \qquad (4)$$

Note that throughout this paper we are assuming that all nodes in the system are "identical", so that the above covariance matrix is independent of n_1 and n_2. Our problem is to find a way of evaluating $\overline{P}_0$ as a function of the dropout process, d_{n_1,n_2}.

Throughout this paper, the dropout process d_{n_1,n_2} will be generated by an N-state Markov chain with states q_1 through q_N, transition probability matrix $Q = [q_{ij}]_{N\times N}$, and stationary distribution $\pi = [\pi_1, \pi_2, \cdots, \pi_N]$. Each node of our system will instantiate a copy of this Markov chain. The state of the Markov chain at node n_2 at time n_1 will be denoted as q_{n_1,n_2}. The Markov chain generates the dropout process through the function h that maps each Markov state, q_i, onto either 0 or 1. The dropout process is then defined by the equation $d_{n_1,n_2} = h(q_{n_1,n_2})$. We can therefore see that the matrix-valued processes A_{n_1,n_2} and B_{n_1,n_2} will take values based on the value of q_{n_1,n_2}. In particular, if $q_{n_1,n_2} = q_i$, then we'll denote A_{n_1,n_2} and B_{n_1,n_2} as A_i and B_i, respectively.

3. PRIOR WORK

There is a small amount of work studying the effect of dropouts on the performance of networked control systems. Nearly all of this work has confined its attention to single control loops, rather than the multiple coupled control loops found in this paper's application. Early work in [8] treated the dropout process as a Markov chain and developed ad hoc schemes for dealing with dropouts. In [12], the system with dropouts was treated as an asynchronous switched system and switched system analysis methods were used to assess overall system stability. Much of the earlier work focused on system stability. More recently, there have been papers examining system *performance* as a function of the dropout process. In [4], this was done for a simple networked control system in which dropped measurements were replaced by zeros. In [10], system performance was measured by its $\mathcal{H}_\infty$ gain and this gain was evaluated as a function of packet loss. Similar results were obtained in [5] and [6] for networked control systems in which the dropout process was modelled as a Markov chain.

The results in [10] and [6] are of particular importance to this paper because they provide formal relationships between system performance and dropout rates. This relationship was used [6] to pose an optimization problem whose solution (if it exists) is a dropout process maximizing control system performance (average output power) subject to a lower bound on the average dropout rate. The top plot in Figure 3 shows results from this prior work. This plot compares the performance of the system under three different stochastic dropout policies; optimal, i.i.d, and a "soft" (m, k)-firm guarantee rule [9] . In this example the system is closed loop stable and open loop unstable (see [6] for details). The Figure plots the average output power as a function of the allowed transmission rate (i.e. 1 minus the average dropout rate). As the transmission rate decreases, the average power increases (worse performance) and asymptotically approaches infinity. The results show that an overload management policy enforcing the optimal dropout process performs better (lower average output signal power over a wider range of dropout rates) than policies enforcing the i.i.d or (m, k)-firm guarantee dropouts processes.

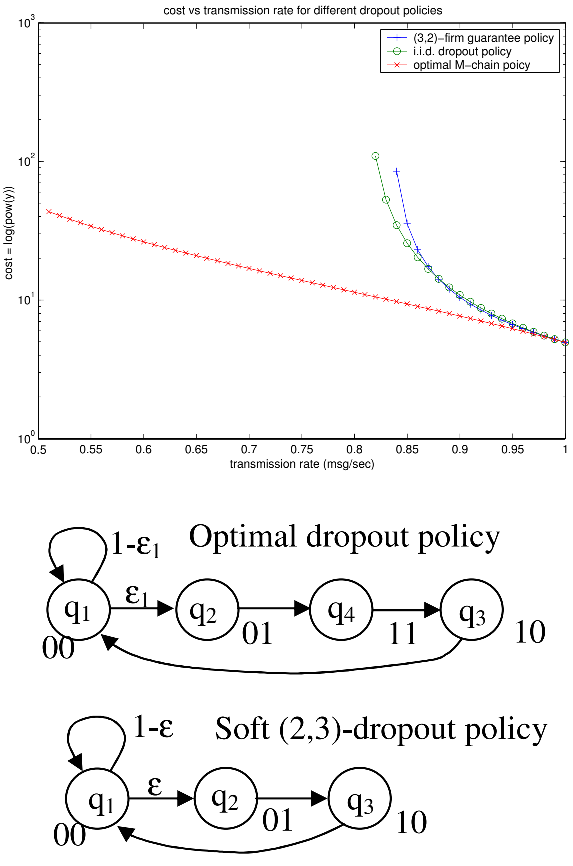


Figure 3: Performance of Various Overload Management Policies

The results in Figure 3 are particularly interesting because the behavior of the optimal dropout process runs counter to prevailing intuition. The bottom pictures in Figure 3 are the state diagrams for the optimal and "soft" $(2, 3)$ dropout processes. The state of each Markov chain is denoted as $d_{n_1-1}d_{n_1}$, so that the state marked "11" means that the last two packets were dropped. Note that the optimal dropout process requires that if a single dropout occurs, we must immediately drop the second packet as well. The soft $(2, 3)$ rule, however, requires that only one packet be dropped in a row. The results show that for this particular system, it is always better to drop two packets than one. This runs directly counter to the prevailing wisdom embodied in the (m, k)-firm guarantee rule. In the (m, k)-firm guarantee rule, we require that m out of k consecutive packets get through the network. Experimental results in [9] showed that overload management policies enforcing this rule perform well on some specific examples. This prior work, however, provides little formal analysis to suggest that this is necessarily the best policy to follow for all control systems. The results in figure 3 show that "soft" variations on the (m, k) heuristic may sometimes perform poorly and show that it is indeed possible to derive overload management policies that optimally utilize the throughput allocated to the controller.

Results of this type are clearly relevant to overload management, for they provide a systematic method by which optimal management policies might be formulated. The results in [10] and [6], however, are not directly applicable to the system shown in Figure 1 because that work only pertains to single control loop whose feedback is implemented over a network. In systems controlled by sensor-actuator networks, there are a large number of control loops that are coupled through their environment. This is precisely the situation encountered in Figure 1. In our system, every node has an embedded processor that implements a local controller. The controller's actions are determined by the neighboring node states and those states are in turn influenced by the local controller through physical interactions in the environment. If we are to develop optimal overload management policies we will need to extend that prior work to this particular class of spatially distributed systems.

4. PERFORMANCE OF SPATIALLY DISTRIBUTED SYSTEMS

This section states two new results concerning the performance of the distributed system in Figure 1. Theorem 4.1 characterizes the average output power $\overline{P}_0$ for the distributed system without control (i.e., $u_{n_1,n_2} = 0$). Theorem 4.2 extends theorem 4.1 to the jump linear systems found in equation 3. This section simply states the theorems and comments on their significance. Formal proofs of the theorems will be found in section 7.

The first theorem characterizes $\overline{P}_0$ for a non-switching spatially distributed control system. In particular, it states that if the system is stable in the mean square sense (i.e. $\mathbf{E}\{x^T_{n_1,n_2}x_{n_1,n_2}\} < \infty$), then $\overline{P}_0$ is obtained from the solution of an infinite system of linear equations.

THEOREM 4.1. *Let x_{n_1,n_2} satisfies equation 1 without control input (i.e., $u_{n_1,n_2} = 0$) where w_{n_1,n_2} is a zero mean white noise process with covariance matrix R.*

If $\mathbf{E}\{x^T_{n_1,n_2}x_{n_1,n_2}\} < \infty$ (i.e. stability in the mean square sense) , then $\overline{P}_0$ (see Eq. 4) is obtained by solving the following system of equations.

$$\begin{aligned}
\overline{P}_0 &= A\overline{P}_0A^T + 2B\overline{P}_0B^T + A(\overline{P}_1 + \overline{P}_1^T)B^T \\
&\quad + B(\overline{P}_1 + \overline{P}_1^T)A^T + B(\overline{P}_2 + \overline{P}_2^T)B^T + R \\
\overline{P}_1 &= A\overline{P}_1A^T + A(\overline{P}_2 + \overline{P}_0)B^T + B(\overline{P}_2 + \overline{P}_0)A^T \\
&\quad + B(\overline{P}_1^T + 2\overline{P}_1 + \overline{P}_3)B^T \\
\overline{P}_k &= A\overline{P}_kA^T + B(\overline{P}_{k-2} + 2\overline{P}_k + \overline{P}_{k+2})B^T \\
&\quad + A(\overline{P}_{k+1} + \overline{P}_{k-1})B^T + B(\overline{P}_{k+1} + \overline{P}_{k-1})A^T
\end{aligned} \tag{5}$$

where the last equation applies for $k \geq 2$.

Remark: The assumption of stability is essential in this theorem. This paper does not determine sufficient conditions for the system to be stable in the mean-square sense, but related results have been obtained in [5] for single networked control loops. Similar stability conditions may be applicable for this distributed systems, but we have not included such results in this paper due to space limitations.

Equations 5 is an infinite system of equations in which you must solve for the matrices $\overline{P}_k$ for $k = 0, \ldots, \infty$. $\overline{P}_0$ was defined in equation 4. The other matrices represent the expectation

$$\begin{aligned} \overline{P}_k &= \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k}\} \qquad (6) \\ \overline{P}_k &= \overline{P}^T_{-k} \end{aligned}$$

Note that because all nodes are identical, this system is shift invariant with respect to both n_1 and n_2. This is why we can drop any explicit mention of n_1 and n_2 in equation 6.

Remark: The infinite set of equations (Eq. 5) may be solved numerically in a recursive manner. In particular, we generate a sequence, $\{\overline{P}_k[L]\}_{L=0}^{\infty}$, of matrices that converge to the true $\overline{P}_k$ as L goes to infinity. The recursion we used is

$$\begin{aligned} \overline{P}_0[L+1] &= A\overline{P}_0[L]A^T + 2B\overline{P}_0[L]B^T + R \qquad (7) \\ &\quad +A(\overline{P}_1[L] + \overline{P}_1[L]^T)B^T + B(\overline{P}_1[L] \\ &\quad +\overline{P}_1[L]^T)A^T + B(\overline{P}_2[L] + \overline{P}_2[L]^T)B^T \\ \overline{P}_1[L+1] &= A\overline{P}_1[L]A^T + A(\overline{P}_2[L] + \overline{P}_0[L])B^T \\ &\quad +B(\overline{P}_2[L] + \overline{P}_0[L])A^T \\ &\quad +B(\overline{P}^T_1[L] + 2\overline{P}_0[L] + \overline{P}_3[L])B^T \\ \overline{P}_k[L+1] &= A\overline{P}_k[L]A^T + A(\overline{P}_{k+1}[L] + \overline{P}_{k-1}[L]B^T \\ &\quad +B(\overline{P}_{k+1}[L] + \overline{P}_{k-1}[L])A^T \\ &\quad +B(\overline{P}_{k-2}[L] + 2\overline{P}_k[L] + \overline{P}_{k+2}[L])B^T \end{aligned}$$

where we let $\overline{P}_0[0] = R$ and $\overline{P}_k[0] = 0$ for $k \neq 0$. The recursion is terminated when $\left\|\overline{P}_0[L+1] - \overline{P}_0[L]\right\|$ is less than a specified error tolerance. This recursion is convergent for this particular example. It is still an open question, however, under what general conditions the proposed recursion is convergent.

We extend Theorem 4.1 to spatially-distributed systems with dropouts. Such systems are modelled as jump linear systems characterized by equation 3. This theorem provides an infinite set of equations that can be used to solve for the covariance matrix,

$$\overline{P}_0 = \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2}\}$$

Once again because of spatial shift-invariance, $\overline{P}_0$ is independent of n_2. The proof for this theorem will be found in section 7.

THEOREM 4.2. *Let w_{n_1,n_2} be a zero-mean white noise process with covariance R. Let x_{n_1,n_2} satisfy the jump linear system equation given in equations 3 which is driven by a Markov chain with transition matrix $Q = [q_{ij}]_{N \times N}$ with stationary distribution $\pi = [\pi_1, \pi_2, \cdots, \pi_N]$.*

If $\mathbf{E}\{x^T_{n_1,n_2} x_{n_1,n_2}\} < \infty$ (i.e. mean square stability) , then

$$\overline{P}_0 = \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2}\} = \sum_{i=1}^{N} \overline{P}_0^{i,i}$$

where $\overline{P}_0^{i,i}$ satisfy the following infinite set of equations.

$$\begin{aligned} \overline{P}_0^{i,i} &= A_i S_0^{i,i} A_i^T + \pi_i B_i (2S_0^{\cdot,\cdot} + S_2^{\cdot,\cdot} + S_{-2}^{\cdot,\cdot}) B_i^T \qquad (8) \\ &\quad +A_i(S_1^{i,\cdot} + S_{-1}^{i,\cdot})B_i^T + B_i(S_1^{\cdot,i} + S_{-1}^{\cdot,i})A_i^T + \pi_i R \\ \overline{P}_1^{i,j} &= B_i(\overline{P}_{-1}^{j,i} + \pi_j S_1^{\cdot,i} + \pi_i \pi_j S_3^{\cdot,\cdot} + \pi_i S_1^{j,\cdot})B_j^T \\ &\quad +\pi_j A_i(S_0^{i,\cdot} + S_2^{i,\cdot})B_j^T + A_i S_1^{i,j} A_j^T \\ &\quad +\pi_i B_i(S_2^{\cdot,j} + S_0^{\cdot,j})A_j^T \\ \overline{P}_k^{i,j} &= A_i S_k^{i,j} A_j^T + \pi_i \pi_j B_i(2S_k^{\cdot,\cdot} + S_{k-2}^{\cdot,\cdot} + S_{k+2}^{\cdot,\cdot})B_j^T \\ &\quad +\pi_j A_i(S_{k+1}^{i,\cdot} + S_{k-1}^{i,\cdot})B_j^T + \pi_i B_i(S_{k+1}^{\cdot,j} + S_{k-1}^{\cdot,j})A_j^T \end{aligned}$$

for $k \geq 2$, where :

$$S_k^{i,j} = \begin{cases} \sum_{l,m=1}^{N} q_{li} q_{mj} \overline{P}_k^{l,m} \ , & \textit{when } k \neq 0 \\ \sum_{l=1}^{N} q_{li} \overline{P}_0^{l,l} \ , & \textit{when k=0 and i=j} \\ 0 \ , & \textit{when k=0 and } i \neq j \end{cases}$$

$$S_k^{i,\cdot} = \sum_{j=1}^{N} S_k^{i,j}, \quad S_k^{\cdot,j} = \sum_{i=1}^{N} S_k^{i,j}, \quad S_k^{\cdot,\cdot} = \sum_{i,j=1}^{N} S_k^{i,j}$$

Remark: Note that there are strong similarities between the equations 8 of theorem 4.2 and equations 5 of theorem 4.1.

Remark: Equations 8 is an infinite set of linear equations that we solve for the matrices $\overline{P}_k^{i,j}$. In particular, these matrices are the following conditional expectations.

$$\begin{aligned} \overline{P}_k^{i,j} &= \pi_i \pi_j \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k} \mid q_{n_1-1,n_2} = q_i, \\ &\qquad q_{n_1-1,n_2-k} = q_j\} \quad \text{for } k \neq 0 \\ \overline{P}_0^{i,i} &= \pi_i \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2} \quad | \quad q_{n_1-1,n_2} = q_i\} \\ \overline{P}_0^{i,j} &= 0 \quad \text{for } i \neq j \\ \overline{P}_{-k}^{i,j} &= \left(\overline{P}_k^{j,i}\right)^T \end{aligned}$$

We can again use a recursion similar to that shown in equation 7 to solve these equations.

5. OPTIMAL OVERLOAD MANAGEMENT POLICIES

Theorem 4.2 presents a method for computing the covariance, $\overline{P}_0$, of a node's state vector as a function of the dropout process' transition matrix, Q. Since we take Trace$(C\overline{P}_0C^T)$ to be a measure of the local controller's performance, these results provide the required extension of [6] to spatially distributed control systems. We can now use this result to formulate and solve a problem to find the Markov chain that minimizes Trace$(C\overline{P}_0C^T)$ subject to a lower bound ε_d on the chain's average dropout rate. The resulting Markov chain will then be used to devise an "optimal" overload management policy.

To state our problem, we need to consider a specific Markov chain. In particular, let's assume the Markov chain has four states defined as follows:

$$\begin{array}{l|l} q_1 & d_{n_1-1,n_2} = 0, d_{n_1,n_2} = 0 \\ q_2 & d_{n_1-1,n_2} = 0, d_{n_1,n_2} = 1 \\ q_3 & d_{n_1-1,n_2} = 1, d_{n_1,n_2} = 0 \\ q_4 & d_{n_1-1,n_2} = 1, d_{n_1,n_2} = 1 \end{array}$$

When $q_{n_1,n_2} = q_i$, then the probability that the next packet will be dropped on purpose is ε_i. In other words ε_i is the probability that the node decided NOT to request data from its neighbors. The total dropout probability (including the effect of link failures) will be $1-(1-\varepsilon_i)(1-f)$ where f is the link failure rate. With these notational conventions, the probability transition matrix for the Markov chain becomes,

$$
\begin{aligned}
Q &= \begin{bmatrix} e_1 & e_2 & 0 & 0 \\ 0 & 0 & e_3 & e_4 \\ e_5 & e_6 & 0 & 0 \\ 0 & 0 & e_7 & e_8 \end{bmatrix} \\
e_1 &= (1-\varepsilon_1)(1-f) \quad e_2 = \varepsilon_1 + f(1-\varepsilon_1) \\
e_3 &= (1-\varepsilon_2)(1-f) \quad e_4 = \varepsilon_2 + f(1-\varepsilon_2) \\
e_5 &= (1-\varepsilon_3)(1-f) \quad e_6 = \varepsilon_3 + f(1-\varepsilon_3) \\
e_7 &= (1-\varepsilon_4)(1-f) \quad e_8 = \varepsilon_4 + f(1-\varepsilon_4)
\end{aligned}
$$

The steady state distribution is denoted as $\pi = [\pi_1, \pi_2, \pi_3, \pi_4]$.

The *average dropout rate*, $\overline{\varepsilon}$ is given by the following equation

$$\overline{\varepsilon} = \sum_{i=1}^{4} \pi_i \varepsilon_i .$$

As discussed earlier, the network will limit a node's transmission rate in order to avoid congestion. This limitation takes the form of a lower bound ε_d on the node's actual dropout rate. In other words, the network requires that the node drops packets at least as fast as ε_d. The overload management policy used by the node must assure that $\overline{\varepsilon} \geq \varepsilon_d$ and that the dropped packets degrade overall control system performance as little as possible. So the optimization problem we seek to solve has the following formal statement,

$$
\begin{array}{ll}
\text{minimize:} & \text{Trace}(\sum_{i=1}^{4} C\overline{P}_0^{i,i} C^T) \\
\text{with respect to:} & \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 \\
\text{subject to:} & \overline{\varepsilon} \geq \varepsilon_d \\
 & \pi = \pi Q \\
 & 1 = \sum_{i=1}^{4} \pi_i
\end{array}
\qquad (9)
$$

where $\overline{P}_0^{i,i}$ is computed using the equations in theorem 4.2 and ε_d is a specified constant (i.e. a measure of the throughput allocated to the local controller).

We considered a specific system to verify the correctness of our theoretical results. This example is a distributed system in which local controllers switch between open-loop and closed-loop configurations. A closed-loop configuration occurs when $d_{n_1,n_2} = 0$. This happens when the Markov chain state, q_{n_1,n_2}, equals q_1 or q_3. An open-loop configuration occurs when a dropout occurs (i.e. $d[n] = 1$). This happens when the Markov chain's state is either q_2 or q_4. The system matrices used in this experiment were,

$$
A = \begin{bmatrix} 0.9990 & 0.0100 \\ -0.1999 & 0.9990 \end{bmatrix}, B = \begin{bmatrix} 0.0005 & 0 \\ 0.1000 & 0 \end{bmatrix},
$$

$$
C = \begin{bmatrix} 1.0 & 0 \\ 0 & 0.1 \end{bmatrix}, F = B.
$$

When there is no dropout, the control u_{n_1,n_2} takes the form

$$u_{n_1,n_2} = Kx_{n_1,n_2} - x_{n_1,n_2-1} - x_{n_1,n_2+1}$$

where $K = \begin{bmatrix} -93.2580 & -10.4700 \\ 0 & 0 \end{bmatrix}$. The dynamics of the closed-loop system therefore become

$$x_{n_1+1,n_2} = (A+BK)x_{n_1,n_2} + Bw_{n_1,n_2},$$

From this equation we see that the closed-loop distributed system is really a group of decoupled subsystems. When a dropout occurs there is no control (i.e. $u_{n_1,n_2} = 0$) and the physical coupling between subsystems reasserts itself. So the system matrices A_i and B_i that are switched to when the Markov chain's state is q_i are given as follows:

$$
\begin{aligned}
A_1 &= A_3 = A + BK \\
B_1 &= B_3 = 0 \\
A_2 &= A_4 = A \\
B_2 &= B_4 = B
\end{aligned}
$$

For our particular problem, `Matlab`'s optimization toolbox was used to numerically solve the preceding optimization problem. In particular, we used the `Matlab` function `fmincon` after a suitable initial condition was identified. These optimizations were done assuming a link failure rate, f, of 0.3 for ε_d between 0.05 and 0.85. The solutions are the transition probabilities ε_i. We found that these probabilities took the following form,

$$
[\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4] = \left\{ \begin{array}{ll} x_1, 0, 0, 0 & \varepsilon_d < 0.2237 \\ 1, 0, x_3, 0 & 0.2237 < \varepsilon_d < 0.4117 \\ 1, x_2, 1, 0 & 0.4117 < \varepsilon_d < 0.5833 \\ 1, 1, 1, x_4 & 0.5833 < \varepsilon_d \end{array} \right.
$$

where x_1, x_2, x_3, x_4 are determined by the average dropout rate condition.

The plot in Figure 4 compares the performance under the "optimal" dropout policy (solid line) and an i.i.d. (independently identically distributed) dropout policy (dashed line). This is the performance computed using our theoretical results. We plot the log power level ($\log(\text{Trace}\mathbf{E}\left[C\overline{P}_0 C\right])$ as a function of the transmission rate $1-\varepsilon_d$. We approximated the infinite distributed system in Figure 1 using a `MatLab SimuLink` model consisting of 27 nodes. All of the simulations assumed free boundary conditions. The results for these simulations are plotted as $*$ and o. We simulated 3 different runs at 6 different dropout rates. In the simulations, we estimated the power by time average taken over 100,000 iterations. The theoretical predictions show that the optimal policy is indeed better than the i.i.d. policy. In reviewing the transition probabilities given above, it is apparent that the optimal policy is a soft $(2,3)$-policy for $\varepsilon_d < 0.2237$. For throughput allocations above this rate, however, the structure of the optimal policy changes to allow higher dropout rates. The simulation results show close agreement with the theoretical predictions for a wide range of dropout rates.

The Markov chains derived in this section form the basis for an overload management policy that is easily implemented on an embedded processor. In particular, a number of these "optimal" Markov chains would be determined for a range of overload conditions (ε_d) and a range of link failure rates (f). We store these transition probabilities in a table that is indexed with respect to ε_d and f. The overload management policy used by each node is a concrete instantiation of the optimal Markov chain whose transition probabilities are loaded from this table based on 1) the throughput (ε_d) that was allocated to the node and 2) based on the link failure rate (f) that was estimated by the node. What should be apparent is that the resulting policy is adaptive with respect to link failure rate and throughput allocation. Moreover, since these chains are solutions to the optimization problem

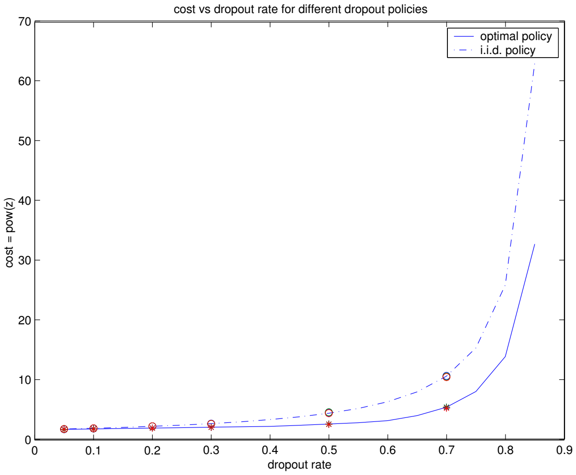


Figure 4: Simulation versus Theoretical Results

in equation 9, we know that this policy degrades application performance as little as possible. In other words, we have provable guarantees that this approach makes optimum use of the throughput allocated to the local controller. The simulation and theoretical results shown in Figure 4 suggest that hardware implementations of such policies should also perform well. This hardware testing, however, remains to be done.

6. FINAL REMARKS

One approach to sensor-actuator network middleware advocates the development of software that is *application independent*. This approach is meaningful for networks that must use a wide variety of traffic types in diverse and unpredictable situations. By their very nature, however, embedded sensor actuator networks are dedicated to a specific type of application whose traffic patterns are often highly regular and predictable. By adopting middleware architectures that ignore our prior knowledge of the plant, we are doomed to under-utilize network resources in support of that application.

This paper demonstrated another approach to network middleware in which application specific knowledge can be used to optimize middleware's support for the application. In particular we were able to use a priori knowledge of the application (in our case a spatially distributed control system) to develop overload management policies that provide provable levels of application performance subject to constraints on the network QoS (as measured by packet dropout rates). The policy developed here is based on a Markov decision process that emerges from solving an optimization problem in which a priori knowledge of the application's dynamics was incorporated. The resulting policy, we feel, can be readily realized as a minor variation on existing overload management schemes, the only real change being that our decision probabilities are based on formal understanding of system dynamics, rather than ad hoc reasoning about the plant.

The preceding discussion focused on sensor-actuator networks in the control of spatially distributed systems. This work, however, is also directly relevant to more traditional sensor networks. In particular, it is possible to apply these results to investigate the impact that dropped sensor measurements would have on the performance of a smoother prediction algorithm used in tracking a vehicle's trajectory through the sensor field. In particular, if we were to use a Kalman filter to smooth sensor measurements in a way that estimates the vehicle's trajectory, then a natural question concerns the impact that missing or intermittent sensor data has on the Kalman filter's performance [11]. In this case the control loop shown in Figure 2 still applies. However, the controller/plant combination is now just the equivalent error dynamics for the Kalman filter, the input disturbance is either sensor or process noise, and the output, y, becomes the state estimation error. This paper's results, therefore, can also be used to characterize the increase in mean state-estimation error as a function of average sensor measurement dropout rate. So while the paper's scope originally rested in distributed control, the results are relevant to pure sensor networks relying on optimal mean square error predictors to track changes in a sensor field.

While this paper's computational results have restricted their attention to spatially invariant and 1-d spatial systems, the analysis could also be extended to 2-d and spatially varying systems. For example, the orignal system equations describe a 2-d nearest neighbor system if we introduce an additional spatial variable n_3 and rewrite equation 1 as

$$\begin{aligned} x_{n_1+1,n_2,n_3} &= Ax_{n_1,n_2,n_3} \\ &\quad +B(x_{n_1,n_2-1,n_3} + x_{n_1,n_2+1,n_3}) \\ &\quad +B(x_{n_1,n_2,n_3-1} + x_{n_1,n_2,n_3+1}) \\ &\quad +F(u_{n_1,n_2,n_3} + w_{n_1,n_2,n_3}) \end{aligned}$$

The analysis inherent in theorem 4.2 assumed spatial invariance. While this assumption simplifies the theorem's proof, a close examination of the proof reveals that the assumption is not necessary for characterizing the covariance matrices. Extending the results in this paper to a spatially varying system is, of course, computationally intensive, but conceptually the extension seems quite straightforward.

To the best of our knowledge, this paper provides one of the few positive examples in which network QoS was related to control system performance in a way that directly leads to practical network protocols. Future work will continue in this direction, by attempting to use formal systems science for the systematic development of embedded network middleware. Over the immediate future we intend to relax some of the assumptions inherent in this paper's development, primarily the reliance on spatial shift-invariance and 1-dimensional structures. Over the long term, we intend to validate the ideas developed in this paper on hardware-in-the-loop experiments.

7. PROOFS

Proof of Theorem 4.1: Let's define the matrix

$$\overline{P}_k = \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k}\}$$

Because all nodes are identical, $\overline{P}_k$ has no dependence on either n_1 or n_2. We also know that

$$\begin{aligned}
x_{n_1+1,n_2} &= Ax_{n_1,n_2} \\
&\quad +B(x_{n_1,n_2-1} + x_{n_1,n_2+1}) \\
&\quad +Fw_{n_1,n_2} \\
x_{n_1+1,n_2-k} &= Ax_{n_1,n_2-k} \\
&\quad +B(x_{n_1,n_2-k-1} + x_{n_1,n_2-k+1}) \\
&\quad +Fw_{n_1,n_2-k}
\end{aligned}$$

We can now expand out $\overline{P}_k$ as

$$\begin{aligned}
\overline{P}_k &= \mathbf{E}\{x_{n_1+1,n_2} x^T_{n_1+1,n_2-k}\} \\
&= A(\overline{P}_{k+1} + \overline{P}_{k-1})B^T + B(\overline{P}_{k+1} + \overline{P}_{k-1})A^T \\
&\quad +A\overline{P}_k A^T + B(\overline{P}_k + \overline{P}_{k-2} + \overline{P}_{k+2})B^T + R_k
\end{aligned}$$

where $R_k = 0$ if $k \neq 0$ and $R_0 = R$ (R is the covariance of the noise process w). The above equation is the third equation in the theorem ($k \geq 2$). We use a similar procedure to get the other two equations in the theorem. •

Proof of Theorem 4.2: Note that if $q_{n_1,n_2} = q_i$ and $q_{n_1,n_2-k} = q_j$, then the states at time n_1+1 can be written as

$$\begin{aligned}
x_{n_1+1,n_2} &= A_i x_{n_1,n_2} \\
&\quad +B_i(x_{n_1,n_2-1} + x_{n_1,n_2+1}) \\
&\quad +Fw_{n_1,n_2} \\
x_{n_1+1,n_2-k} &= A_j x_{n_1,n_2-k} \\
&\quad +B_j(x_{n_1,n_2-k-1} + x_{n_1,n_2-k+1}) \\
&\quad +Fw_{n_1,n_2-k} \qquad (10)
\end{aligned}$$

Here, when $q_{n_1,n_2} = q_i$, we denote $A(q_{n_1,n_2})$ and $B(q_{n_1,n_2})$ as A_i and B_i respectively.

So we can use this to write out for $k \geq 2$,

$$\begin{aligned}
\overline{P}^{i,j}_k &= \pi_i\pi_j \mathbf{E}\{x_{n_1+1,n_2} x^T_{n_1+1,n_2-k} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\} \\
&= \pi_i\pi_j A_i \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\} A_j^T \\
&+ \pi_i\pi_j A_i (\mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k-1} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\} \\
&\quad +\mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k+1} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\}) B_j^T \\
&+ \pi_i\pi_j B_i (\mathbf{E}\{x_{n_1,n_2-1} x^T_{n_1,n_2-k} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\} \\
&\quad +\mathbf{E}\{x_{n_1,n_2+1} x^T_{n_1,n_2-k} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\}) A_j^T \\
&+ \pi_i\pi_j B_i (\mathbf{E}\{x_{n_1,n_2-1} x^T_{n_1,n_2-k-1} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\} \\
&\quad +\mathbf{E}\{x_{n_1,n_2-1} x^T_{n_1,n_2-k+1} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\} \\
&\quad +\mathbf{E}\{x_{n_1,n_2+1} x^T_{n_1,n_2-k-1} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\} \\
&\quad +\mathbf{E}\{x_{n_1,n_2+1} x^T_{n_1,n_2-k+1} \mid q_{n_1,n_2} = q_i, \\
&\quad q_{n_1,n_2-k} = q_j\}) B_j^T
\end{aligned}$$

There are nine conditional expectations in the above equation. The first expectation can be simplified as follows,

$$\begin{aligned}
\pi_i\pi_j \quad & \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k} | q_{n_1,n_2} = q_i, \\
& q_{n_1,n_2-k} = q_j\} \\
= \quad & \sum_{l,m=1}^{N} \pi_i\pi_j \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k} \mid q_{n_1,n_2} = q_i, \\
& q_{n_1,n_2-k} = q_j, q_{n_1-1,n_2} = q_l, q_{n_1-1,n_2-k} = q_m\} \\
& P(q_{n_1-1,n_2} = q_l, q_{n_1-1,n_2-k} = q_m \mid \\
& q_{n_1,n_2} = q_i, q_{n_1,n_2-k} = q_j) \\
= \quad & \sum_{l,m=1}^{N} \pi_i\pi_j \mathbf{E}\{x_{n_1,n_2} x^T_{n_1,n_2-k} \mid q_{n_1-1,n_2} = q_l, \\
& q_{n_1-1,n_2-k} = q_m\} P(q_{n_1-1,n_2} = q_l \mid q_{n_1,n_2} = q_i) \cdot \\
& P(q_{n_1-1,n_2-k} = q_m \mid q_{n_1,n_2-k} = q_j) \\
= \quad & \pi_i\pi_j \sum_{l=1}^{N} \sum_{l,m} \overline{P}^{l,m}_k \frac{q_{mj} q_{li}}{\pi_i\pi_j} \\
= \quad & \sum_{l=1}^{N} \sum_{m=1}^{N} q_{li} q_{mj} \overline{P}_{l,m}(L)
\end{aligned}$$

The second expectation can be simplified as shown below. The third, fourth, and seventh expectations have similar derivations and aren't shown.

$$\begin{aligned}
\pi_i\pi_j \quad & \mathbf{E}\{x_{n_1,n_2}x^T_{n_1,n_2-k-1} \mid q_{n_1,n_2} = q_i, \\
& q_{n_1,n_2-k} = q_j\} \\
= \quad & \pi_i\pi_j\mathbf{E}\{x_{n_1,n_2}x^T_{n_1,n_2-k-1} \mid q_{n_1,n_2} = q_i\} \\
= \quad & \pi_i\pi_j \sum_{l,m=1}^{N} \mathbf{E}\{x_{n_1,n_2}x^T_{n_1,n_2-k-1} \mid q_{n_1,n_2} = q_i, \\
& q_{n_1-1,n_2} = q_l, q_{n_1-1,n_2-k-1} = q_m\} \\
& P(q_{n_1-1,n_2} = q_l, q_{n_1-1,n_2-k-1} = q_m \mid q_{n_1,n_2} = q_i) \\
= \quad & \pi_i\pi_j \sum_{l,m=1}^{N} \mathbf{E}\{x_{n_1,n_2}x^T_{n_1,n_2-k-1} \mid q_{n_1-1,n_2} = q_l, \\
& q_{n_1-1,n_2-k-1} = q_m\} \\
& P(q_{n_1-1,n_2} = q_l \mid q_{n_1,n_2} = q_i) \\
& P(q_{n_1-1,n_2-k-1} = q_m) \\
= \quad & \pi_j \sum_{l,m=1}^{N} q_{li}\overline{P}^{l,m}_{k+1}
\end{aligned}$$

The fifth expectation can be simplified as shown below: The sixth, eight, and ninth expectations have similar deriviations and aren't shown,

$$\begin{aligned}
\pi_i\pi_j \quad & \mathbf{E}\{x_{n_1,n_2-1}x^T_{n_1,n_2-k-1}|q_{n_1,n_2} = q_i, \\
& q_{n_1,n_2-k} = q_j\} \\
= \quad & \pi_i\pi_j\mathbf{E}\{x_{n_1,n_2-1}x^T_{n_1,n_2-k-1}\} \\
= \quad & \pi_i\pi_j \sum_{l,m=1}^{N} \overline{P}^{l,m}_{k}
\end{aligned}$$

With these simplifications inserted into equation 10, we obtain the third equation in theorem 4.2. A similar derivation can be used to obtain the first two equations. •

8. REFERENCES

[1] R. D'Andrea (1999). Linear Matrix Inequalities, Multidimensional System Optimization, and Control of Spatially Distributed Systems: An Example. In Proceedings, American Control Conference, pages 2713-2717, 1999.

[2] R. D'Andrea and G. E. Dullerud (2000). Distributed Control of Spatially Interconnected Systems. Accepted for publication, IEEE Transactions on Automatic Control.

[3] Frampton, K.D (2001), Decentralized Control of Structural Acoustic Radiation, Proceedings of Symposium on Noise Control and Acoustics IMECE 2001, Noise Control and Acoustics Division, NY, NY, November 2001

[4] C.N. Hadjicostis and R. Touri (2002), "Feedback control utilizing packet dropping network links", IEEE Conference on Decision and Control, Las Vegas, USA, Dec. 2002.

[5] Q. Ling and M.D. Lemmon (2002), "Robust Performance of Soft Real-time Networked Control Systems with Data Dropouts", IEEE Conference on Decision and Control, Las Vegas, 2002.

[6] Q. Ling and M.D. Lemmon (2003), Soft real-time scheduling of networked control systems with dropouts governed by a Markov chain, American Control Conference, Denver, Colorado, June 2003.

[7] M. Mariton (1990), *Jump Linear Systems in Automatic Control*, Marcel Dekker Inc., 1990.

[8] J. Nilsson (1998), *Real-time control systems with delays*, Ph.D. thesis, Lund Institute of Technology, 1998.

[9] P. Ramanathan (1999), "Overload management in real-time control applications using (m, k)-firm guarantee", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 10, no. 6, pp. 549-559, June 1999.

[10] P. Seiler (2001), "Coordinated Control of Unmanned Aerial Vehicles", Ph.D. Thesis, University of California Berkeley, Mechanical Engineering, Fall 2001.

[11] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry (2003), Kalman Filter with Intermittent Observation , to appear in the 42nd IEEE Conference on Decision and Control, Hawaii, Dec. 9-12, 2003.

[12] W. Zhang (2001), *Stability analysis of networked control systems*, Ph.D. Thesis, Case Western Reserve University, 2001.

On Routing in Random Rayleigh Fading Networks

Martin Haenggi, *Senior Member, IEEE*

Abstract

This article addresses the routing problem for large wireless networks of randomly distributed nodes with Rayleigh fading channels. First, we establish that the distances between neighboring nodes in a Poisson point process follow a generalized Rayleigh distribution. Based on this result it is then shown that, given an end-to-end packet delivery probability (as a QoS requirement), the energy benefits of routing over many short hops are significantly smaller than for deterministic network models that are based on the geometric disk abstraction. If the permissible delay for short-hop routing and long-hop routing is the same, it turns out that routing over fewer but longer hops may even outperform nearest-neighbor routing, in particular for high end-to-end delivery probabilities.

Index terms – Communication systems, routing, fading channels, ad hoc networks, probability, Poisson processes.

1 Introduction

Energy consumption in multi-hop wireless networks is a crucial issue that needs to be addressed at all the layers of the communication system, from the hardware up to the application. In this article, we focus on routing strategies that employ hops of different length in a network whose nodes constitute a Poisson point process. The analysis is based on a Rayleigh fading channel model, and the results demonstrate that the properties of the physical channel have a substantial impact on optimum protocol design at the network layer.

Often, a deterministic "disk model" is used for the analysis of multi-hop packet networks [1–9], where the radius for a successful transmission has a deterministic value, irrespective of the condition of the wireless channel. Interference is taken account using the same geometric disk abstraction. The stochastic nature of the fading channel and thus the fact that the signal-to-noise-and-interference ratio (SINR) is a random variable are neglected. Using such models, it is easy to show that, for a path loss exponent of α, there is an energy gain of $n^{\alpha-1}$ if a hop over a distance d is split into n hops of distance d/n. However, the volatility of the channel cannot be ignored in wireless networks [10, 11]; the inaccuracy of "disk models" has also been pointed out in [12] and is easily demonstrated experimentally [13, 14]. In addition the "prevalent all-or-nothing model" [15] leads to the assumption that a transmission over a multi-hop path either fails completely or is 100% successful, ignoring the fact that end-to-end packet loss probabilities increase with the number of hops (unless the transmit power is adapted).

To overcome some of these limitations of the "disk model", we employ a simple Rayleigh fading link model that relates transmit power, large-scale path loss, and the success of a transmission. The end-to-end packet delivery probability over a multi-hop route is the product of the link-level reception probabilities.

While fading has been considered in the context of packet networks [16, 17], its impact on the network (and higher) layers is largely an open problem. This paper attempts to shed some light on this cross-layer issue by analyzing the performance of different routing schemes under Rayleigh fading. In Section 2, we introduce the link model and show that noise and interference can be treated separately. We also prove that the internode distances in a Poisson point process follow a generalized Rayleigh distribution; we define and determine the path efficiency; and we introduce the five routing strategies we consider. In Section 3, the energy consumption of those five routing strategies is analyzed; Section 4 discusses the performance of those routing schemes under equal delay constraints, and Section 5 concludes the paper.

2 The Rayleigh Network Model

2.1 The Rayleigh fading link model

We assume a narrowband Rayleigh block fading channel. A transmission from node i to node j is successful if the SINR γ_{ij} is above a certain threshold Θ that is determined by the communication hardware, and the modulation and coding scheme [10]. The SINR γ is a discrete random process given by

$$\gamma = \frac{R}{N+I} \, . \tag{1}$$

R is the received power, which is exponentially distributed with mean $\bar{R}$. Over a transmission of distance $d = \|x_i - x_j\|_2$ with an attenuation d^α, we have $\bar{R} = P_0 d^{-\alpha}$, where P_0 is proportional to the transmit power[1]. N denotes the noise power, and I is the interference power affecting the transmission, *i.e.*, the sum of the received power from all the undesired transmitters.

Theorem 1 *In a Rayleigh fading network, the reception probability $p_r := \mathbb{P}[\gamma \geqslant \Theta]$ can be factorized into the reception probability of a zero-noise network and the reception probability of a zero-interference network.*

Proof: Let R_0 denote the received power from the desired source and R_i, $i = 1, \ldots, k$, the received power from k interferers. All the received powers are exponentially distributed, *i.e.*, $p_{R_i}(r_i) = 1/\bar{R}_i \, e^{-r_i/\bar{R}_i}$, where $\bar{R}_i$ denotes the average received power $\bar{R}_i = P_i d_i^{-\alpha}$. The probability of correct reception is[2]

$$\begin{aligned} p_r =& \mathbb{P}[R_0 \geqslant \Theta(I+N)] && (2)\\ =& \mathbb{E}_I\Big[\exp\Big(-\frac{\Theta(I+N)}{\bar{R}_0}\Big)\Big] && (3)\\ =& \int_0^\infty \cdots \int_0^\infty \exp\Big(-\frac{\Theta(\sum_{i=1}^k r_i + N)}{\bar{R}_0}\Big) \prod_{i=1}^k p_{R_i}(r_i)\,\mathrm{d}r_1 \cdots \mathrm{d}r_k && (4)\\ =& \underbrace{\exp\Big(-\frac{\Theta N}{P_0 d_0^{-\alpha}}\Big)}_{p_r^N} \cdot \underbrace{\prod_{i=1}^k \frac{1}{1+\Theta \frac{P_i}{P_0}\big(\frac{d_0}{d_i}\big)^\alpha}}_{p_r^I} \,. && (5) \end{aligned}$$

p_r^N is the probability that the SNR $\gamma_N := R_0/N$ is above the threshold Θ, *i.e.*, the reception probability in a zero-interference network as it depends only on the noise. The second factor p_r^I is the reception probability in a zero-noise network. □

This allows an independent analysis of noise and interference. If the load is light, then the number of interferers k is relatively small and/or their distances are relatively big, which implies SIR≫SNR, thus the noise analysis alone provides accurate results. This has been demonstrated in [18] in the case of slotted ALOHA with transmit probability p. Indeed, for small p, the throughput is proportional to p, indicating that there are no losses due to collisions, *i.e.*, $p_r^I \approx 1$ for all timeslots and all transmissions. For high load, a separate

[1] This equation does not hold for very small distances. So, a more accurate model would be $\bar{R} = P_0' \cdot (d/d_0)^{-\alpha}$, valid for $d \geqslant d_0$, with P_0' as the average value at the reference point d_0, which should be in the far field of the transmit antenna. At 916MHz, for example, the near field may extend up to 1m (several wavelengths).

[2] A similar calculation has been carried out in the Appendix of [16] for a network with spreading gain and equal transmit powers for all nodes.

interference analysis has to be carried out to determine p_r^I. Note that *power scaling*, *i.e.*, scaling the transmit powers of all the nodes by the same factor, does not change the SIR (p_r^I only depends on *power ratios*), but (slightly) increases the SINR. Since only p_r^N is affected by absolute energy levels, we will focus on zero-interference networks to identify energy-efficient routing strategies.

In a zero-interference network, the reception probability over a link of distance d at a transmit power P_0, is given by $p_r := \mathbb{P}[\gamma_N \geqslant \Theta] = e^{-\frac{\Theta N}{P_0 d^{-\alpha}}}$, therefore

$$P_0 = \frac{d^\alpha \Theta N}{-\ln p_r} \,. \tag{6}$$

Note that for high probabilities, the packet loss probability $1 - p_r$ is tightly upperbounded by the normalized mean NSR $\Theta N/\bar{R}_0$=$\Theta/\bar{\gamma}_N$ [19]. Since $-\ln p_r \approx 1 - p_r$, we can also say that the packet loss probability is inversely proportional to the transmit power for high p_r.

2.2 Poisson random networks

We consider a Poisson point process of intensity λ in the plane, where the probability of finding k nodes in an area A is given by the Poisson distribution[3]

$$\mathbb{P}[k \text{ nodes in } A] = e^{-\lambda A} \frac{(\lambda A)^k}{k!} \,. \tag{7}$$

Note that for infinite networks, the Poisson point process corresponds to a uniform distribution [12, 20], and for large networks, the two distributions are equivalent for all practical purposes. Henceforth we denote a network whose nodes constitute a two-dimensional Poisson point process as a *Poisson random network*. Without loss of generality, we will restrict ourselves to the case $\lambda = 1$ (unit density), since the product λA can always be scaled such that $\lambda = 1$.

For the routing schemes we consider, we need to determine the distance from one node to its neighboring nodes that lie within a sector ϕ, *i.e.*, within $\pm\phi/2$ of the source-destination axis (Fig. 1).

Proposition 1 (Distance to nearest neighbor.) *In a Poisson random network with unit density, the distance R between a node and its nearest neighbor in a sector ϕ is Rayleigh distributed with mean $\sqrt{\pi/(2\phi)}$.*

[3] This can be generalized to higher dimensions if A is the Lebesgue measure of the subset considered.

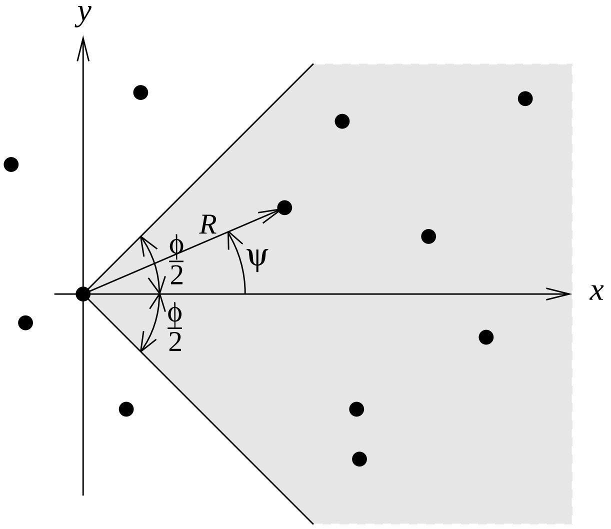


Figure 1: Part of a Rayleigh network with the source at the origin and the x-axis pointing towards the destination node. R denotes the distance to the nearest neighbor within a sector ϕ around x, and ψ is its argument. Hence (R, ψ) are the polar coordinates of the nearest neighbor within a sector ϕ, and (X, Y) are its Cartesian coordinates.

Proof: Let R be the distance to the nearest neighbor in a sector ϕ. The probability that there is no neighbor in a sector ϕ up to a distance r is the complementary cumulative distribution $\mathbb{P}[R > r] = e^{-r^2\phi/2}$, thus the probability density is $p_R(r) = r\phi\, e^{-r^2\phi/2}$, which is a Rayleigh distribution with mean $\sqrt{\pi/(2\phi)}$ and variance $2/\phi - \pi/(2\phi) = (4-\pi)/(2\phi)$. The distribution of the argument ψ is uniform between $-\phi/2$ and $\phi/2$. □

Definition 1 (Rayleigh network.) *A* Rayleigh network *is a Poisson random network where the physical channel is subject to Rayleigh fading.*

To compare different routing schemes, we need to define the *path efficiency.*

Definition 2 (Path efficiency.) *The* path efficiency *is the ratio of the Euclidean distance between the end nodes and the actually travelled distance.*

As an example, the path efficiency (for long connections[4]) in a square lattice network with nearest-neighbor routing is

$$\kappa(\phi) = \frac{1}{|\cos\phi| + |\sin\phi|} \tag{8}$$

[4] This is based on the assumption that the angle towards the destination is uniformly distributed over $[0, 2\pi)$.

where ϕ denotes the direction from transmitter to receiver relative to the orientation of the grid. The maximum is 1, the minimum is $1/\sqrt{2}$ (at $\phi = \pi/4 + k\pi/2$), and the expected value is

$$\eta := \mathbb{E}[\kappa] = \frac{2\sqrt{2}}{\pi} \operatorname{arctanh} \frac{\sqrt{2}}{2} \approx 0.79\,. \tag{9}$$

We will use η to denote the expected path efficiency.

Definition 3 (Link efficiency.) *The* link efficiency *is $X/R = \cos(\Psi)$, where X is the x-coordinate of the position of the nearest neighbor (see Fig. 1).*

In Rayleigh networks with long connections, the distances and angles to nearest neighbors are iid, so the expected path efficiency equals the expected link efficiency:

Proposition 2 (Path efficiency in Rayleigh networks.) *In a Rayleigh network with nearest-neighbor routing in a sector ϕ, the expected path efficiency for a long connection is*

$$\eta(\phi) = \frac{2}{\phi} \sin\left(\frac{\phi}{2}\right) \approx 1 - \frac{\phi^2}{24}\,, \tag{10}$$

where the approximation is the second-order Taylor expansion.

Proof: The expected link efficiency is $\mathbb{E}[\cos\Psi]$, and since the hop distances are iid, this is also the expected path efficiency. Thus we have

$$\eta(\phi) = \mathbb{E}[\cos\Psi] = \frac{2}{\phi} \int_0^{\phi/2} \cos\psi \, d\psi = \frac{2}{\phi} \sin\left(\frac{\phi}{2}\right)\,. \tag{11}$$

□

To compare the *transport capacity* of different routing schemes, we have to determine the *progress* X. Changing to Cartesian coordinates, the joint probability $p_{XY}(x,y)$ for the nearest neighbor's position (in a sector $\phi \leqslant \pi$) is

$$p_{XY}(x,y) = e^{-\frac{x^2+y^2}{2}\phi}\,, \qquad x \geqslant 0, \quad -x\tan(\phi/2) \leqslant y \leqslant x\tan(\phi/2) \tag{12}$$

Integration with respect to y yields the marginal density

$$p_X(x) = \operatorname{erf}\left(\sqrt{\frac{\phi}{2}} \tan\left(\frac{\phi}{2}\right) x\right) \sqrt{\frac{2\pi}{\phi}}\, e^{-x^2\phi/2}\,, \tag{13}$$

where $\operatorname{erf}(\cdot)$ denotes the error function, *i.e.*, $\operatorname{erf}(x) := 2/\sqrt{\pi} \int_0^x e^{-t^2} dt$. It is easily verified that for small ϕ, as $X \approx R$, this tends to a Rayleigh distribution. On the other hand, for $\phi \to \pi$, $\tan(\phi/2) \to \infty$, and the error function tends to 1, and the distribution approximates the one-sided Gaussian $\sqrt{2}e^{-x^2\pi/2}$, $x \geqslant 0$, with mean $\sqrt{2}/\pi$. Y has zero mean and becomes Gaussian for $\phi \to \pi$ with variance $1/\pi$. Note that X and Y are not independent unless $\phi = \pi$ or $\phi = 2\pi$.

The *expected progress* $\mathbb{E}[X]$ is

$$\mathbb{E}[X] = \sqrt{\frac{\pi}{2\phi}}\,\frac{2}{\phi}\sin\left(\frac{\phi}{2}\right)\,, \tag{14}$$

which corresponds, as expected, to the product of the mean path efficiency $\eta(\phi)$ (10) and the mean of the Rayleigh distribution $\sqrt{\pi/(2\phi)}$.

Next, we generalize Proposition 1.

Proposition 3 (Distance to n-th neighbor.) *The probability density of the distance to the n-th neighbor in a sector ϕ is*

$$p_{R_n}(r) = r^{2n-1}\left(\frac{\phi}{2}\right)^n \frac{2}{(n-1)!}\,e^{-r^2\phi/2} \tag{15}$$

Proof: Let S_k be the k-th coefficient in the Poisson distribution: $S_k := (r^2\phi/2)^k/k!$. The probability that there are less than n nodes closer than r in the sector ϕ is

$$P_n := \mathbb{P}[0\ldots n{-}1 \text{ nodes within } r] = \sum_{k=0}^{n-1} S_k\,e^{-r^2\phi/2}\,. \tag{16}$$

Hence,

$$p_{R_n} = \frac{d}{dr}\,(1-P_n) = \left(r\phi\sum_{k=0}^{n-1} S_k - \sum_{k=1}^{n-1}\underbrace{\frac{k(r^2\phi/2)^{k-1}}{k!}}_{S_{k-1}}\,r\phi\right)e^{-r^2\phi/2}\,. \tag{17}$$

The only term that is not cancelled in the two sums is the one at $n-1$, leading to

$$p_{R_n} = r\phi\,\cdot\,\underbrace{S_{n-1}\,e^{-r^2\phi/2}}_{\text{Erlang distribution}}\,, \tag{18}$$

which is identical to (15). □

Since p_{R_n} is a Rayleigh distribution for $n = 1$, it can be considered a generalized Rayleigh distribution. Similarly, in one dimension, the Erlang distribution is a generalized exponential distribution. So, the transition from one dimension to two dimensions simply entails a multiplication by $r\phi$ (that comes from the inner derivative of the exponential part) in the distributions of the node distances.

The mean of R_n is given by

$$\mathbb{E}[R_n] = \frac{\sqrt{2}}{\sqrt{\phi}}\,\frac{\Gamma(n+1/2)}{\Gamma(n)} \approx \frac{\sqrt{2}}{\sqrt{\phi}}\sqrt{n-1+\frac{\pi}{4}}\,. \tag{19}$$

To get this approximation, we take the first term from the series expansion $\frac{\Gamma(n+1/2)}{\Gamma(n)} = \sqrt{n}(1 - \frac{1}{8n} + O(\frac{1}{n^2}))$ [21][5] and, noting that this is not accurate for small n, adjust it (by adding $\pi/4 - 1$ to n) such that the approximation is precise for $n = 1$. For $n > 1$, this yields a very tight upper bound.

The second moment is $2n/\phi$, hence the variance is

$$\mathrm{Var}[R_n] = \frac{2n}{\phi} - E[R_n]^2 \approx \frac{4-\pi}{2\phi}\,, \tag{20}$$

using the same approximation. From the series expansion above, we know that $\mathrm{Var}[R_n] = \frac{1}{2\phi} + O(\frac{1}{n})$. So we have found sharp lower and upper bounds on the variance:

$$\frac{4-\pi}{2\phi} \leqslant \mathrm{Var}[R_n] < \frac{1}{2\phi} \quad \forall n \in \mathbb{N} \tag{21}$$

Since $4 - \pi \approx 1$, we can conclude that, interestingly, the variance is almost independent of n[6].

2.3 Energy consumption of a route

Assume a connection from node 0 to node n in a multi-hop network. The desired end-to-end reliability is P_{EE}. The reception probability of a chain of n nodes is

$$p_n = \prod_{i=1}^{n} e^{-\Theta/\bar{\gamma}_i} = e^{-\Theta \sum_{i=1}^{n} \frac{1}{\bar{\gamma}_i}} \tag{22}$$

where $\bar{\gamma}_i$ denotes the mean SNR at link i.

Assume that the maximum transmit power $P_{\max}$ is sufficient to reach node n in a single hop with probability of P_{EE}. The question is how many hops are optimum in terms of energy consumption. First, we consider the case of a one-dimensional chain of equidistant nodes with distance d. Let E_0 be the energy required for a transmission over distance d with probability P_{EE}, *i.e.*, $E_0 := -d^\alpha \Theta N / \ln P_{EE}$. Covering the total distance in one single hop requires an energy of $E_1 = n^\alpha E_0$. In the multi-hop case with n hops, a reception probability $p_r = \sqrt[n]{P_{EE}}$ is required at each hop. Since $\ln P_{EE} = n \ln p_r$, the total energy in this case is $E_n = n \cdot n\, E_0$. So, for $\alpha = 2$, there is no benefit in multi-hop routing.

[5] Note that this series can also be derived by using the identity $\frac{\Gamma(n+1/2)}{\Gamma(n)} = \frac{(2n)!\sqrt{\pi}}{n!(n-1)!4^n}$ [22] and applying Stirling's approximation.

[6] This is not the case for one-dimensional networks, where the variance increases linearly in n (variance of the Erlang distribution with parameter n).

For a regular rectangular lattice network where the maximum-hop case corresponds to routing over nearest neighbors, we still have $E_n = n^2 E_0$, whereas for the single-hop case, we get $E_1 = (n\kappa)^\alpha E_0$, where κ is the path efficiency. The energy consumption is identical for

$$\alpha = \frac{2 \log n}{\log(n\kappa)} \,. \tag{23}$$

For $\alpha = 4$ and $\kappa = 1/\sqrt{2}$, we get $E_1 = E_2$, so even for large path loss exponents, there is no benefit in dividing a one-hop diagonal connection into two nearest-neighbor hops.

2.4 Routing strategies

Here, we introduce the five routing strategies that will be analyzed in the next Section. In order to implement those routing algorithms, it is assumed that all nodes know their own location and that the source node knows the direction towards the destination. The strategies are the following (see Fig. 2):

A. Multi-hop: Transmit hop by hop over n nearest-neighbor hops in a sector ϕ, *i.e.*, the next-hop node is the nearest neighbor that lies within $\pm\phi/2$ of the axis to the destination.

B. Single-hop: Transmit directly to the n-th node in the route found in A.

C. Single-hop: Transmit directly to the n'-th nearest neighbor in the sector ϕ.

D. Single-hop: Transmit directly to the n-th nearest neighbor in a sector $\phi' < \phi$.

E. Single-hop: Transmit directly to the nearest neighbor in a sector $\phi'' < \phi'$.

Note that "single-hop" here does not mean that the (final) destination is reached in one hop, but it means that a certain progress X_0 is made in one hop rather than in n hops. So, relative to the end-to-end connection, all the schemes are multi-hop, but A uses short hops, whereas B through E use long(er) hops. For a fair comparison, we shall choose n', ϕ', and ϕ'' for strategies C, D and E, respectively, such that the expected progress $\bar{X}$ is the same as for A and B. As long as routing decisions are taken locally, virtually all routing strategies fall into those categories; whatever the specific properties of an algorithm are, the goal will always be to make some progress in the right direction.

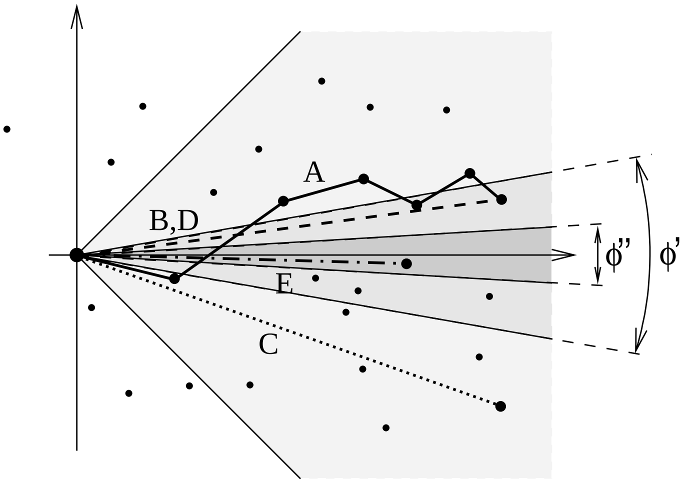


Figure 2: Illustration of the different routing strategies. Strategie A follows the solid line (multiple hops), B and D follow the dashed line, C follows the dotted line, and E follows the dash-dotted line. The sectors ϕ, ϕ', and ϕ'' are represented by different grades of gray.

3 Energy Consumption

In this Section, the energies required to deliver a packet over a progress $\bar{X} := n\eta(\phi)\mathbb{E}[R(\phi)] = n\eta(\phi)\sqrt{\pi/(2\phi)}$ with probability P_{EE} are determined for those five strategies, normalized by $\frac{\Theta N}{-\ln P_{EE}}$.

3.1 Analysis of the five routing strategies

Strategy A. With $p_r = \sqrt[n]{P_{EE}}$, the total expected energy consumption is $n^2\mathbb{E}[R^\alpha]$:

$$E_A = n^2 \underbrace{\left(\frac{2}{\phi}\right)^{\alpha/2} \Gamma\left(1+\frac{\alpha}{2}\right)}_{\mathbb{E}[R^\alpha]} \tag{24}$$

Strategy B. First, we need to establish the path efficiency. In strategy A, the angle to the next node is uniformly distributed in $\pm\phi/2$. With increasing n, the argument Ψ from the origin to the n-th node tends to be Gaussian distributed with variance $V(\phi, n)$. Since the support of the pdf of Ψ is always $[-\phi/2, \phi/2]$[7], the variance decreases inversely proportional

[7] Hence, after every convolution, the support needs to be scaled, which results in a reduction of the variance.

to n, *i.e.*, $V(\phi,n) \approx \frac{\phi^2}{12n}$. So, for large n, we get[8]

$$\begin{aligned}\eta_B(\phi,n) &= \mathbb{E}[\cos\Psi] && (25)\\ &\approx \frac{\sqrt{6n}}{\phi\sqrt{\pi}} \cdot \int_{-\infty}^{\infty} \cos\psi\, e^{-6n\left(\frac{\psi}{\phi}\right)^2} \mathrm{d}\psi && (26)\\ &= e^{-\frac{\phi^2}{24n}} && (27)\\ &\approx 1 - \frac{\phi^2}{24n}\,, && (28)\end{aligned}$$

from which we find $1/\eta_B(n,\phi)^\alpha \approx 1 + \frac{\alpha\phi^2}{24n}$. Clearly, $\eta_B(n,\phi) \geqslant \eta(\phi)$ with equality only for $n = 1$. Since we are transmitting over a distance $nR\frac{\eta}{\eta_B}$, we get

$$E_B = n^\alpha \mathbb{E}[R^\alpha]\Big(\frac{\eta}{\eta_B}\Big)^\alpha \approx n^\alpha \left(\frac{2}{\phi}\right)^{\alpha/2} \Gamma\left(1+\frac{\alpha}{2}\right)\left(1 - \frac{\alpha\phi^2(n-1)}{24n}\right) \qquad (29)$$

So, the increase of $n^{\alpha-2}$ compared with strategy A is partially compensated for by the factor $(\eta/\eta_B)^\alpha$, *i.e.*, the increase in path efficiency.

Strategy C. A direct transmission to the n'-th neighbor in the sector ϕ shall result in the same expected progress as in strategy A, *i.e.*, $\mathbb{E}[R_{n'}] = n\mathbb{E}[R]$. From (19), we have

$$\frac{\sqrt{2}}{\sqrt{\phi}}\sqrt{n' - 1 + \frac{\pi}{4}} = n\sqrt{\frac{\pi}{2\phi}}\,, \qquad (30)$$

which yields

$$n' = \frac{\pi}{4}(n^2 - 1) + 1\,, \qquad (31)$$

independent of ϕ. The path efficiency for this single-hop transmission is still $\eta(\phi)$, since the argument of the destination node is uniformly distributed in the sector ϕ. We have[9]

$$\mathbb{E}[R^\alpha_{n'}] = \left(\frac{2}{\phi}\right)^{\alpha/2} \frac{\Gamma\left(n' + \frac{\alpha}{2}\right)}{\Gamma(n')} \approx \left(\frac{2}{\phi}\right)^{\alpha/2} \left(n' - 1 + \sqrt{\alpha}\frac{\pi}{4}\right)^{\alpha/2}\,. \qquad (32)$$

Plugging in (31) gives

$$\mathbb{E}[R^\alpha_{n'}] \approx \left(\frac{\pi}{2\phi}\right)^{\alpha/2} \left(n^2 + \sqrt{\alpha} - 1\right)^{\alpha/2}. \qquad (33)$$

[8] The Gaussian approximation is very accurate even for small n. For $\phi = \pi/2$ and $n = 1$, *e.g.*, the precise value is $2\sqrt{2}/\pi \approx 0.9003$, whereas this approximation yields $e^{-\pi^2/96} \approx 0.9023$, so the error is only 0.2%. The second order Taylor expansions are identical, even for $n = 1$.

[9] Again, we use the series expansion $\frac{\Gamma(n'+\alpha/2)}{\Gamma(n')} = n'^{\frac{\alpha}{2}}(1 - O(\frac{1}{n'}))$ and adjust the constant such that the approximation is precise for $\alpha = n' = 1$. Note that for $\alpha = 1$, this is identical to (19). For $\alpha = 2$, $\mathbb{E}[R^\alpha_{n'}] \propto n'$, and for $\alpha = 4$, $\mathbb{E}[R^\alpha_{n'}] \propto n'^2$. The absolute error in the approximation for $\alpha = 2$ is $\pi\sqrt{2}/4 - 1 \approx 0.11$, independent of n'.

Strategy D. The reduction of the sector angle from ϕ to ϕ' shall ensure that the transmission to the n-th nearest neighbor within ϕ' results in the same progress as n nearest-neighbor hops in ϕ, *i.e.*,

$$n\mathbb{E}[R(\phi)] \cdot \eta(\phi) = \mathbb{E}[R_n(\phi')] \cdot \eta(\phi') \,. \tag{34}$$

Since R and R_n depend on ϕ and ϕ', respectively, ϕ' cannot be expressed in a closed form. However, we can find a tight upper bound on the energy consumption (lower bound on ϕ') by assuming that the path efficiency does not increase with decreasing angle:

$$\phi' \gtrapprox \phi \, \frac{4(n-1)+\pi}{n^2\pi} \,. \tag{35}$$

$\phi' \approx \phi/n$ with high accuracy. With (32), we get the upper bound B_D

$$E_D := \mathbb{E}[R_n^\alpha] \lessapprox B_D := \left(\frac{2\pi n^2}{\phi(4(n-1)+\pi)}\right)^{\alpha/2} \frac{\Gamma(n+\frac{\alpha}{2})}{\Gamma(n)} \,. \tag{36}$$

To get a lower bound, we assume that the path efficiency for ϕ' is 1. Solving $n\mathbb{E}[R(\phi)]\eta(\phi) = \mathbb{E}[R_n(\phi')]$ using (19) yields

$$\phi' \lessapprox \phi \, \frac{4(n-1)+\pi}{n^2\pi} \frac{1}{\eta(\phi)^2} \,, \tag{37}$$

and we find

$$B_D \gtrapprox E_D \gtrapprox B_D \eta(\phi)^\alpha \,. \tag{38}$$

For large n, E_D tends to the lower bound. Since B_D has the same asymptotic behavior for large n as strategy C, strategy D outperforms C.

Strategy E. Here we determine an even smaller angle ϕ'' such that the first neighbor within that sector has the same expected distance as n hops in the original sector ϕ. As in D, the equation $n\mathbb{E}[R(\phi)]\eta(\phi) = \mathbb{E}[R(\phi'')]\eta(\phi'')$ does not have a closed-form solution, and we have to bound E_E. From $\eta(\phi'') > \eta(\phi)$ we have $\phi'' \gtrapprox \phi/n^2$ and thus

$$E_E < B_E := n^\alpha \left(\frac{2}{\phi}\right)^{\alpha/2} \Gamma\left(1+\frac{\alpha}{2}\right) \,. \tag{39}$$

On the other hand, assuming $\eta(\phi'') = 1$, we find a tight (for $n > 2$) lower bound $B_E\eta(\phi)^\alpha$, hence

$$B_E > E_E \gtrapprox B_E \eta(\phi)^\alpha \,. \tag{40}$$

Strategy	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$
A	$\frac{4}{\pi}\frac{1}{\eta^2} \approx \frac{4}{\pi}\big(1+\frac{\phi^2}{12}\big)$	$\frac{6}{\pi}\frac{1}{\eta^3}\frac{1}{n} \approx \frac{6}{n\pi}\big(1+\frac{\phi^2}{8}\big)$	$\frac{32}{\pi^2}\frac{1}{\eta^4}\frac{1}{n^2} \approx \frac{32}{n^2\pi^2}\big(1+\frac{\phi^2}{6}\big)$
B	$\frac{4}{\pi}\frac{1}{\eta_B^2} \approx \frac{4}{\pi}\big(1+\frac{\phi^2}{12n}\big)$	$\frac{6}{\pi}\frac{1}{\eta_B^3} \approx \frac{6}{\pi}\big(1+\frac{\phi^2}{8n}\big)$	$\frac{32}{\pi^2}\frac{1}{\eta_B^4} \approx \frac{32}{\pi^2}\big(1+\frac{\phi^2}{6n}\big)$
C	$\left(1+\frac{\frac{4}{\pi}-1}{n^2}\right)\frac{1}{\eta^2}$	$\approx \left(1+\frac{(\sqrt{3}-1)^{\frac{3}{2}}}{n^2}\right)\frac{1}{\eta^3}$	$\left(1+\frac{\frac{12}{\pi}-2}{n^2}+\frac{\frac{32}{\pi^2}-\frac{12}{\pi}+1}{n^4}\right)\frac{1}{\eta^4}$
D	$\approx \left(1+\frac{\frac{4}{\pi}-1}{n}\right)\big(1+\frac{\phi^2}{12n^2}\big)$	$\approx \left(1+\frac{\frac{6}{\pi}-1}{n}\right)\big(1+\frac{\phi^2}{8n^2}\big)$	$\approx \left(1+\frac{\frac{32}{\pi^2}-1}{n}\right)\big(1+\frac{\phi^2}{6n^2}\big)$
E	$\frac{4}{\pi}\frac{1}{\eta''^2} \approx \frac{4}{\pi}\big(1+\frac{\phi^2}{12n^4}\big)$	$\frac{6}{\pi}\frac{1}{\eta''^3} \approx \frac{6}{\pi}\big(1+\frac{\phi^2}{8n^4}\big)$	$\frac{32}{\pi^2}\frac{1}{\eta''^4} \approx \frac{32}{\pi^2}\big(1+\frac{\phi^2}{6n^4}\big)$

Table 1: Normalized energy consumption of the five routing strategies. The approximations are second-order Taylor expansions.

3.2 Comparison

In the previous Subsection, the energy was calculated for a transmission over a distance of $\bar{X} = n\mathbb{E}[R]\eta$. To determine the energy required *per unit distance*, the energies are divided by $\bar{X}^\alpha$. Table 1 compares the five proposed routing strategies.
Remarks:

- For strategy A, the energy consumption decreases with increasing n for $\alpha > 2$. If we take end-to-end probabilities and path efficiencies into account, the benefit of multihop routing under Rayleigh fading is substantially smaller than under the deterministic "disk model": for $\alpha = 2$, it is always detrimental to use short hops, and for $\alpha = 3$, $\phi = \pi/2$ and $n = 2$, we have $E_B/E_A \approx 1.76$, so skipping one hop is only 76% more expensive, rather than $2^{\alpha-1} = 400\%$, as an overly simplistic analysis would yield (see the remarks in Section 1).

- The single-hop strategies B-E all have comparable performance. With increasing n, their energy consumption decreases since the path efficiency increases.

- For strategy C, and $\alpha = 3$, the approximation does not precisely agree with A for $n = 1$ due to the approximation of the $\Gamma(\cdot)$ function for half-integers.

- For strategies B,D, and E, the path efficiencies tend to 1 with increasing n at different speeds.

- Strategy D is the only one where the energy consumption goes to 1 with increasing n.

In terms of the complexity of the implementation of those schemes, B and D are similar to A, while E is simpler, since only one nearest neighbor has to be identified. C, on the other hand, is impractical for large n due to the increase in n^2 in (31). For $n = 5$, for example, the 20th neighbor would need to be identified, which may cause too much overhead.

3.3 Determination of the sector angle ϕ

From the energy analysis, it can be seen that it is desirable to choose ϕ as small as possible to achieve a high path efficiency. On the other hand, we have to make sure that a sufficient number of neighbors can be reached with high probability within the sector. The maximum distance $\hat{R}$ that can be covered at the peak transmit power $P_{\max}$ to guarantee a reception probability p_r is

$$\hat{R} = \left(\frac{P_{\max}(-\ln p_r)}{\Theta N}\right)^{\frac{1}{\alpha}} = \left(\frac{\gamma_{\max}}{\Theta}(-\ln p_r)\right)^{\frac{1}{\alpha}} . \tag{41}$$

If p_c is the desired probability of finding a neighbor within ϕ, we get from Proposition 1

$$\phi \geqslant \frac{-2\ln(1-p_c)}{\hat{R}^2} = -2\ln(1-p_c)\left(\frac{\Theta N}{P_{\max}\ln p_r}\right)^{\frac{2}{\alpha}} . \tag{42}$$

For $\phi = \pi/2$, for example, the expected distance to the nearest neighbor is 1, but for $p_c = 99.9\%$ (high probability of connectivity), a range $\hat{R} = \sqrt{12\ln 10/\pi} \approx 3$ is needed.

4 Delay Considerations

Clearly, if the delay constraints are tight, routing over many short hops may be prohibitively slow. If, on the other hand, the delay induced by an n-hop connection is tolerable, the energy consumption in the single-hop case can be further reduced by exploiting time diversity in the form of retransmissions.

4.1 Multiple long-hop transmissions

For a fair comparison, we assume a delay constraint of n timeslots for both the multi-hop (over n hops, *i.e.*, one transmission in every slot) and the single-hop schemes.

We consider two strategies to exploit the fact that n timeslots are available in the case of a single-hop transmission (over an n times larger distance), namely the schemes with and without availability of channel state information (CSI) at the transmitter.

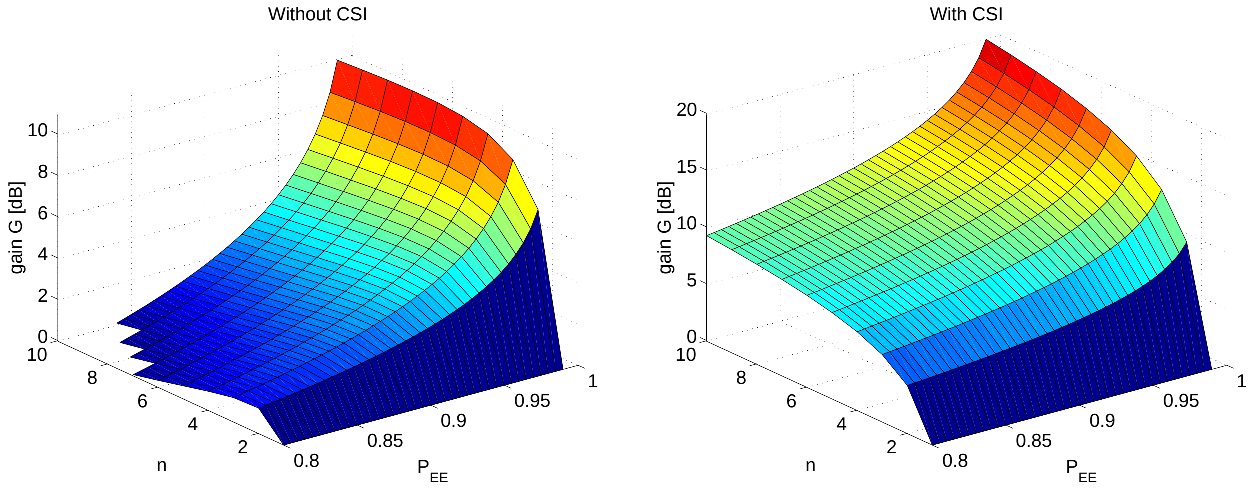


Figure 3: Gain of the n-transmission schemes without (left) and with (right) channel state information over a single transmission in the first timeslot as a function of the number of available time slots n and the desired reliability P_{EE}. Points with gain smaller than 1 are not shown.

No CSI: Transmit *n* times. For a block fading channel, we can assume independent fading levels at each time slot. Since the delay constraint gives time for (at most) n transmissions, there is a benefit from time diversity (selection combining). The required single-use reception probability $P_{EE,1}$ is then given by

$$P_{EE,1} = 1 - (1 - P_{EE})^{\frac{1}{n}} \,. \tag{43}$$

Compared with the single-transmission case, this leads to an energy gain of

$$G = \frac{\log P_{EE,1}}{n \, \log P_{EE}} \,, \tag{44}$$

since from (6), the necessary transmit power in the single-transmission case is $-\frac{\Theta N d^\alpha}{\log P_{EE}}$, whereas for the multi-transmission case, it is $-\frac{n \Theta N d^\alpha}{\log P_{EE,1}}$.

If we assume a simple ARQ scheme, where a short ACK or NACK packet is sent immediately after the receipt of the data packet, this gain is almost doubled (and the average delay cut in half), since after a successful transmission, the packet does not have to be retransmitted any longer.

With CSI: A single opportunistic transmission when the channel is best. With CSI at the transmitter, the packet can be scheduled to be transmitted when the fading

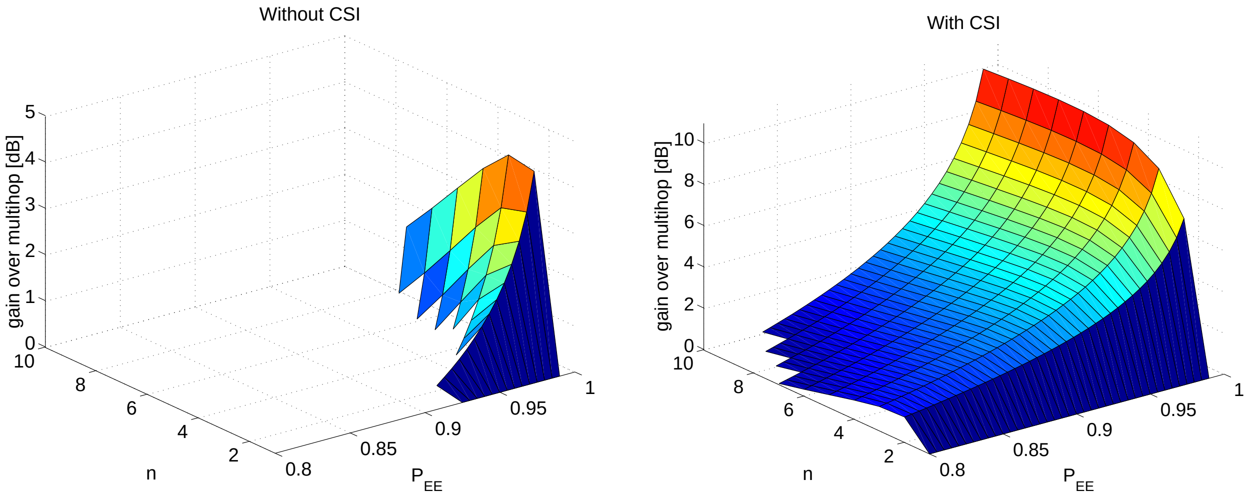


Figure 4: Gain of the schemes without (left) and with (right) channel state information over a multi-hop scheme as a function of the number of available time slots n and the desired reliability P_{EE}. The path loss exponent α is 3. Points with gain smaller than 1 are not shown.

level is best. In other words, the best time slot out of n is picked for a single transmission of the packet. Note that if the delay constraint n is not hard but an average constraint, then this scheme may choose from $2n$ slots. The cumulative distribution of the maximum of n fading random variables is $F_n(x) = (1 - e^{-x/\bar{R}})^n$ with $\bar{R} = P_0 d^{-\alpha}$. From

$$P_{EE} = 1 - \left(1 - e^{-\frac{\Theta N}{\bar{R}}}\right)^n \qquad \Longrightarrow \qquad \bar{R} = \frac{\Theta N}{-\ln\left(1 - (1 - P_{EE})^{\frac{1}{n}}\right)}\,, \tag{45}$$

we see that the reception probability increases in the same way as in the multiple transmission case but with only one transmission. So, this strategy has an n times higher gain than the one without CSI.

Figure 3 shows the gains for both strategies (as a function of the number of transmissions n and the desired probability) over a single-transmission scheme. For small P_{EE} and large n, the gain drops below 1 for the scheme without CSI (see the left plot).

Figure 4 compares the single-hop retransmission-based strategies (n transmissions) with conventional multi-hop routing over n hops for $\alpha = 3$. It can be seen that the single-hop schemes often outperform multi-hop routing *if the same delay is permitted.* Since the factor n between the strategies with and without CSI corresponds to an increase of the path loss exponent by one, we find in general

$$G_{\mathrm{CSI}}(\alpha) = G_{\mathrm{NCSI}}(\alpha - 1)\,. \tag{46}$$

So, the left plot in Fig. 4 shows the gain G_{CSI} for $\alpha = 4$.
Note that in this analysis, we did not take into account the higher path efficiency of the single-hop scheme. So, the effective gain is slightly bigger.

4.2 Multiple transmissions in the short-hop case

If the delay constraints are not tight, then there is time for retransmissions even for the short-hop routing scheme. Indeed, most wireless link layer protocols are based on ARQ, *i.e.*, acknowledgments and retransmissions are employed to ensure a satisfactory reliability. A given desired link reliability P_L can be achieved by a single transmission or multiple transmissions at lower power. The following Proposition addresses the question which scheme is energy-optimum.

Proposition 4 (Optimum power level.) *The energy-optimum transmit power level P_0^{opt} to achieve a link reliability $P_L > 1/2$ is given by*

$$P_0^{opt} = \frac{\Theta N d^\alpha}{\ln 2}\,, \tag{47}$$

the corresponding single-transmission reception probability is $p_r^{opt} = 1/2$, and the necessary average number of transmissions is[10] $n_t = -\log_2(1 - P_L)$.

Proof: The average number of transmissions to achieve P_L is given by

$$n_t = \begin{cases} \frac{\log(1-P_L)}{\log(1-p_r)} & p_r < P_L \\ 1 & p_r \geqslant P_L \end{cases}\,. \tag{48}$$

So, the total energy consumption is

$$E(P_0) = n_t\, P_0 = \frac{\log(1 - P_L)}{\log\left(1 - e^{-\frac{\Theta N}{P_0 d^{-\alpha}}}\right)}\, P_0\,, \tag{49}$$

which is minimized at

$$P_0^{\text{opt}} = \arg\min_{P_0} E(P_0) = \frac{\Theta N d^\alpha}{\ln 2}\,, \tag{50}$$

corresponding to $p_r^{\text{opt}} = 1/2$, and, in turn, $n_t = -\log_2(1 - P_L)$. □

[10] $\log_2(\cdot)$ is the binary logarithm, while $\log(\cdot)$ denotes any logarithm.

Note that the optimum does not depend on P_L. The factor $1/\ln 2 \approx 1.44$ corresponds to 1.6dB, hence the transmit power that minimizes the overall energy consumption is 1.6dB above $\Theta N d^\alpha$. The minimum energy value is then given by

$$E(P_0^{\text{opt}}) = \Theta N d^\alpha \, \frac{-\ln(1-P_L)}{(\ln 2)^2} \,. \tag{51}$$

With $E(P_0^{\text{single}}) = -\frac{d^\alpha \Theta N}{\ln P_L}$, the energy gain is

$$\frac{E(P_0^{\text{opt}})}{E(P_0^{\text{single}})} = \log_2(1-P_L)\,\log_2 P_L\,, \qquad P_L > \frac{1}{2}\,. \tag{52}$$

The reduction in transmit power is $-\ln 2/\ln P_L$. For $P_L \approx 1$, this is approximately $\ln 2/(1-P_L)$ or $\ln 2$ divided by the desired packet loss rate. The gain then is inversely proportional to the packet loss rate times the logarithm of the packet loss rate. For $P_L = 99\%$, for example, the reduction in transmit power is $\approx 100 \ln 2 \approx 69$, the gain is slightly bigger than 10dB, and $n_t \approx 6.64$. So, the reduction in transmit power more than compensates for the higher number of transmissions. Of course, the single-transmission scheme and the energy-optimum scheme are just the two extreme possibilities in the energy-delay trade-off.

Again, to be fair, the long-hop scheme should be permitted the same delay. For the case without CSI, if the short-hop scheme routes over n hops with an average of n_t transmissions at each hop, the long-hop scheme may use nn_t timeslots, *i.e.*, it may transmit nn_t times instead of just n times. The required reliability is $P_{EE} = P_L^n$, so we find that the energy consumption for n transmissions is

$$E_n = n \frac{\Theta N d'^\alpha}{-\ln(1-(1-P_{EE})^{\frac{1}{n}})}\,, \tag{53}$$

where d' denotes the distance over this long hop. Permitting nn_t transmissions at lower transmit power, we have

$$E_{n\,n_t} = nn_t \frac{\Theta N d'^\alpha}{-\ln(1-(1-P_{EE})^{\frac{1}{nn_t}})}\,, \tag{54}$$

which results in an energy gain of

$$G = \frac{E_n}{E_{n\,n_t}} = \frac{1}{n_t}\frac{\log(1-(1-P_{EE})^{\frac{1}{nn_t}})}{\log(1-(1-P_{EE})^{\frac{1}{n}})}\,. \tag{55}$$

Since we know that the optimum single-transmission reception probability is 1/2, this gain is smaller than 1 if $1-(1-P_{EE})^{\frac{1}{n}} \leqslant 1/2$. Therefore, only for high P_{EE} or small n, there is a benefit in transmitting $n\,n_t$ times. In contrast, the scheme with CSI available at the transmitter always benefits from a larger number of timeslots.

In summary, for *very* loose delay constraints, short-hop schemes derive more benefit from retransmissions. Since there is an optimum number of transmissions, the long-hop scheme cannot benefit from having a very large number of timeslots available with this simple retransmission scheme (unless CSI is available). In this case, the available time should rather be exploited by using a stronger channel code, thereby lowering the SINR threshold Θ.

5 Concluding Remarks

We have offered a fresh look at the routing problem in random wireless networks by taking into account Rayleigh fading, end-to-end packet delivery probabilities, node distributions, and path efficiencies. The main result is that, from an energy point of view, routing over many short hops is not as beneficial in a Rayleigh network as it seems to be for the deterministic "disk model". If we impose a delay constraint that is, for a fair comparison, loose enough such that it can be met by short-hop routing, long-hop schemes may actually *save* energy.

Apart from energy and delay, there are other important practical issues that determine the overall performance of a routing scheme:

Interference. The different transmit power levels of the different routing schemes will change the interference by which other ongoing transmissions will be affected. However, if all nodes use a long-hop strategy rather than a short-hop strategy, their relative power levels will stay the same, and the interference part in the reception probability (5), p_r^I, remains constant. Even if an individual source node chooses to adopt a long-hop strategy, the total transmission time will be shorter, which may offset the increase in power. In general, it is not clear whether multiple short hops or a single long hop causes more interference [10]. Total radiated energy may be an indicator.

Diversity schemes. Path and cooperative diversity have been shown to enhance the performance of wireless networks [19,23]. In nearest-neighbor routing, there is little room to exploit cooperative strategies. If longer hops are used, a number of intermediate nodes may overhear the transmission and assist by relaying the packet if needed (provided that they are not asleep). In [4], the number of disjoint and independent paths are investigated. It has been shown that long-hop routing does not decrease the number of independent paths but increases the number of disjoint paths.

Receiver power consumption and sleep modes. The energy consumption for packet reception may be substantial. So, nodes that are not part of an active route (or cooperative scheme) should be put to sleep. Long-hop routing permits a more aggressive use of sleep modes than short-hop routing.

Amplifier characteristics. If the efficiency of the power amplifier depends on the radiated power, then the routing scheme should ensure that the PA mostly operates close to its saturated power $P_{\max}$ [24]. In this case, the variance of the distances becomes important, and nearest-neighbor routing should be avoided, as pointed out in [25].

Route acquisition time. Here, the long-hop schemes have a clear advantage, which is particularly important for on-demand routing algorithms [26].

Route breaks due to mobility and node failure. With less relaying nodes, the probability that new routes have to be discovered due to failing or moving nodes is smaller. Hence the *robustness* of the network is bigger. The availability of intermediate nodes also increases the network's capability of route repair and maintenance.

Energy and delay balancing. Often, the data gathered by a sensor networks has to be delivered to a single point, the *observer* or *base station.* Using nearest-neighbor routing, the lifetime of a node close to the base station is much shorter than the lifetime of a node far away due to traffic accumulation [27]. With long-hop routing, the energy consumption is better balanced. Instead of just a few critical nodes, there are many more, which results in an increase of the network lifetime.

In conclusion, long-hop strategies have a clear advantage in most of these categories, in addition to being competitive in terms of energy.

Acknowledgments

The partial support of the DARPA/IXO-NEST Program (AF-F30602-01-2-0526), NSF (ECS02-25265 and ECS03-29766), and ORAU is gratefully acknowledged. The author would also like to thank the anonymous reviewers for their helpful comments.

References

[1] J. A. Silvester and L. Kleinrock, “On the Capacity of Multihop Slotted ALOHA Networks with Regular Structure,” *IEEE Transactions on Communications*, vol. COM-31, pp. 974–982, Aug. 1983.

[2] H. Takagi and L. Kleinrock, “Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals,” *IEEE Transactions on Communications*, vol. COM-32, pp. 246–257, Mar. 1984.

[3] L. Hu, “Topology Control for Multihop Packet Networks,” *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1474–1481, 1993.

[4] J. L. Wang and J. A. Silvester, “Maximum Number of Independent Paths and Radio Connectivity,” *IEEE Transactions on Communications*, vol. 41, pp. 1482–1493, Oct. 1993.

[5] M. Sanchez, P. Manzoni, and Z. Haas, “Determination of Critical Transmission Range in Ad-Hoc Networks,” in *Multiaccess, Mobility and Teletraffic for Wireless Communications (MMT’99)*, (Venice, Italy), Oct. 1999.

[6] P. Gupta and P. R. Kumar, “The Capacity of Wireless Networks,” *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, Mar. 2000.

[7] M. Grossglauser and D. Tse, “Mobility Increases the Capacity of Ad-hoc Wireless Networks,” in *IEEE INFOCOM*, (Anchorage, AL), 2001.

[8] G. Németh, Z. R. Turányi, and A. Valkó, “Throughput of Ideally Routed Wireless Ad Hoc Networks,” *ACM Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 40–46, 2001.

[9] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Optimizing Sensor Networks in the Energy-Latency-Density Design Space,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, 2002.

[10] A. Ephremides, “Energy Concerns in Wireless Networks,” *IEEE Wireless Communications*, vol. 9, pp. 48–59, Aug. 2002.

[11] A. J. Goldsmith and S. B. Wicker, “Design Challenges for Energy-Constrained Ad Hoc Wireless Networks,” *IEEE Wireless Communications*, vol. 9, pp. 8–27, Aug. 2002.

[12] E. S. Sousa and J. A. Silvester, “Optimum Transmission Ranges in a Direct-Sequence Spread-Spectrum Multihop Packet Radio Network,” *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 762–771, June 1990.

[13] D. A. Maltz, J. Broch, and D. B. Johnson, “Lessons from a Full-Scale Multihop Wireless Ad Hoc Network Testbed,” *IEEE Personal Communications*, vol. 8, pp. 8–15, Feb. 2001.

[14] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An Empirical Study of Epidemic Algorithms in Large Scale Multihop Wireless Networks," 2002. Intel Research Report IRB-TR-02-003. Available at `http://www.intel-research.net/Publications/Berkeley/050220021703_19.pdf`.

[15] T. J. Shepard, "A Channel Access Scheme for Large Dense Packet Radio Networks," in *ACM SIGCOMM*, (Stanford, CA), Aug. 1996. Available at: `http://www.acm.org/sigcomm/sigcomm96/papers/shepard.ps`.

[16] M. Zorzi and S. Pupolin, "Optimum Transmission Ranges in Multihop Packet Radio Networks in the Presence of Fading," *IEEE Transactions on Communications*, vol. 43, pp. 2201–2205, July 1995.

[17] Y. Y. Kim and S. Li, "Modeling Multipath Fading Channel Dynamics for Packet Data Performance Analysis," *Wireless Networks*, vol. 6, pp. 481–492, 2000.

[18] M. Haenggi, "Probabilistic Analysis of a Simple MAC Scheme for Ad Hoc Wireless Networks," in *IEEE CAS Workshop on Wireless Communications and Networking*, (Pasadena, CA), Sept. 2002.

[19] M. Haenggi, "Analysis and Design of Diversity Schemes for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 19–27, Jan. 2005. Available at `http://www.nd.edu/~mhaenggi/pubs/jsac_adhoc.pdf`.

[20] R. Meester and R. Roy, *Continuum Percolation.* Cambridge University Press, 1996.

[21] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science, 2nd ed.* Reading, MA: Addison-Wesley, 1994.

[22] G. Arfken, *Mathematical Methods for Physicists.* Orlando, FL: Academic Press, 3rd ed., 1985.

[23] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Transactions on Information Theory*, vol. 50, pp. 3062–3080, Dec. 2004. Available at: `http://www.nd.edu/~jnl/pubs/it2002.pdf`.

[24] S. C. Cripps, *RF Power Amplifiers for Wireless Communications.* Artech House, 1999. ISBN 0-89006-989-1.

[25] M. Haenggi, "Twelve Reasons not to Route over Many Short Hops," in *IEEE Vehicular Technology Conference (VTC'04 Fall)*, (Los Angeles, CA), Sept. 2004. Available at `http://www.nd.edu/~mhaenggi/pubs/vtc04.pdf`.

[26] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," *IEEE Personal Communications*, vol. 6, pp. 46–55, Apr. 1999.

[27] M. Haenggi, "Energy-Balancing Strategies for Wireless Sensor Networks," in *IEEE International Symposium on Circuits and Systems (ISCAS'03)*, (Bangkok, Thailand), May 2003. Available at `http://www.nd.edu/~mhaenggi/pubs/iscas03.pdf`.

Energy-Balancing Strategies for Wireless Sensor Networks

Martin Haenggi

Dept. of Electrical Engineering

University of Notre Dame

Notre Dame, IN 46556, USA

mhaenggi@nd.edu

Abstract

The lifetime of wireless sensor network is crucial, since autonomous operation must be guaranteed over an extended period. Usually all the sensor data has to be forwarded to an observer or base station via multihop routing, which causes the traffic pattern in the network to be highly nonuniform and puts a high burden on the sensor nodes close to the observer. We propose and analyze five strategies that balance the energy consumption of the nodes to increase the lifetime of the network substantially: distance variation with and without rate adaptation, compression, routing, and reliability balancing. Since several of these schemes affect the transmission delay significantly, remarks on their delay performance are also given. The analyses are based on a non-fading channel model with additive white Gaussian noise (AWGN) and a Rayleigh fading model.

Keywords — Sensor networks, radio communication, AWGN channels, packet switching.

1 Introduction

Large-scale networks of integrated wireless sensors become increasingly tractable, as advances in hardware technology and engineering design have led to dramatic reductions in size, power consumption, and cost for digital circuitry, wireless communications, and MEMS. This enables very compact and autonomous nodes, each containing one or more sensors, computation and communication capabilities, and a power supply. Multihop routing is typically used to reduce the transmit power and, consequently, increase the battery lifetime and decrease the interference between the nodes, thereby allowing spatial reuse of the communication channel.

Wireless sensor networks [1, 2] differ from other types of multihop wireless networks by the fact that, in most cases, the sensor data has to be delivered to a single sink, the *observer* or *base station* (BS). Clearly, one of the primary concerns is the lifetime of the network. Although different definitions of lifetime exist [3], a sensor network certainly has to be considered "dead" whenever it is no longer able to forward any data to the BS. We assume that every sensor node in the network has an equal probability of generating data packets that are forwarded to the BS via multihop routing using other sensor nodes as relays. Apparently, the burden on the nodes close to the base station is considerably higher than on the nodes that are far away. Figure 1 depicts a possible arrangement of sensor nodes and identifies the most critical nodes in the network. Without appropriate measures, they will die quickly, rendering the network useless. This is a crucial problem in sensor networking that has received surprisingly little attention.

While sensor networks are the most prominent examples for such many-to-one traffic, there are two other classes of networks where the same problems occurs:

- **Multihop Extension of Wireless Local Area Networks (WLANs).** To extend the coverage are of the WLAN, every node also acts as a router, forwarding packets on behalf of other nodes that are further away from the BS (or access point or gateway) [4]. To account

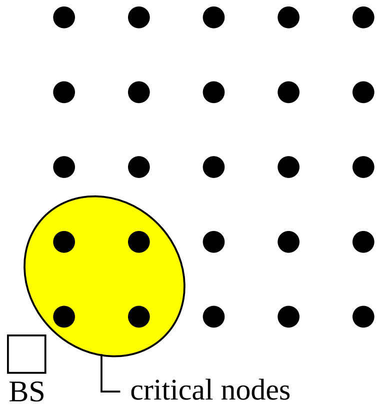


Figure 1: A sensor network with a base station.

for user mobility, the network is self-organizing and self-configuring, similarly to a pure ad hoc network. In [5], the authors show that the per-node capacity decrases in proportion to the number of nodes, but they do not address energy issues.

- **Multihop Cellular Networks.** In [6], a new architecture, the multihop cellular network (MCN) has been proposed as a viable alternative to the conventional single-hop cellular network (SCN) by combining the features of SCNs and ad hoc networks. The BS in MCN is acting as a special relay or gateway to support inter-cell communications or connections to the backbone telephone network. While [6] permitted direct communication between mobile stations (MSs) within a cell, this was excluded in subsequent work [7–10], where all MSs are restricted to communicate with the BS only, *i.e.*, a multihop many-to-one or one-to-many scheme. In other words, the traffic of one MS can be forwarded by other MSs to reach the BS.

In this paper, we propose and discuss strategies to ensure maximum lifetime of the network by balancing the energy load as equally as possible.

The analysis is based on channel models with and without Rayleigh fading and AWGN (additive white Gaussion noise). Such models characterize the wireless link more accurately than the "disk model" that is often used, where it is assumed that the radius for a successful transmission of a packet has a fixed and deterministic value, irrespective of the condition and realization of the wireless channel [11–14]. Such simplified link models ignore the probabilistic nature of the wireless channel and the fact that the *signal-to-noise-and-interference ratio*, that determines the success of a transmission, is a random variable. The volatility of the channel cannot be ignored in wireless networks [3, 15]; the inaccuracy of "disk models" has also been pointed out in [16] and is easily demonstrated experimentally [17, 18]. The (non-fading) AWGN model used in this paper relates the signal-to-noise ratio (SNR) and the capacity of a channel, whereas the Rayleigh fading model relates the average SNR with the success of a transmission at a given rate. The end-to-end packet delivery probability over a multihop route is the product of the link-level reception probabilities.

2 Link Models

2.1 AWGN channels

The complex AWGN channel has a capacity $C = \log_2(1+\gamma)$, where $\gamma = LP_0 d^{-\alpha}/N_0$ is the signal-to-noise ratio (SNR), P_0 the transmit power, d the link distance, α the path loss exponent, N_0 the noise variance, and L some constant that depends on the antenna gains and wavelength. For simplicity, we will henceforth assume $L = 1$. In the presence of normally distributed interference, the SNR can be augmented to a signal-to-noise-and-interference ratio (SINR) by adding the interference to the noise term. We assume that the channel code is capacity-achieving, *i.e.*, that we can transmit reliably at rate $R = C$. Note that for a practical channel code with finite block length, we necessarily have $R < C$ and/or error-prone transmissions; however, this loss in throughput (or the additional energy expenditure) does not change our comparative results. Given R, the transmit power is $P_0 = (2^R - 1)N_0 d^\alpha$. If the packet transmission time is normalized to unity, the power equals the transmission energy. We will denote by E_0 the energy required to transmit one packet at rate R

over a unit distance, *i.e.*, $E_0 = (2^R - 1)N_0$. In general, $E = d^\alpha E_0$.

2.2 Rayleigh fading channels

This model is based on a flat Rayleigh block fading channel [19]. The transmission is successful if the SINR γ at the receiver is above a certain threshold Θ that is determined by the transmission rate, the communication hardware, and the modulation and coding scheme (normally between 1 and 100 or 0dB and 20dB). In any case, $\Theta > 2^R - 1$ due to capacity and rate considerations. With the assumptions above, the received power Q is exponentially distributed with mean $\bar{Q} = P_0 d^{-\alpha}$.

The following theorem proves useful for the analysis:

Theorem 1 *In a Rayleigh fading network, the reception probability* $p_r = \mathbb{P}[\gamma \geqslant \Theta]$ *can be factorized into the reception probability of a zero-interference network* p_r^N *and the reception probability of a zero-noise network* p_r^I *as follows:*

$$p_r = \underbrace{\exp\Big(-\frac{\Theta N_0}{P_0 d_0^{-\alpha}}\Big)}_{p_r^N} \cdot \underbrace{\prod_{i=1}^{k} \frac{1}{1+\Theta\frac{P_i}{P_0}\big(\frac{d_0}{d_i}\big)^\alpha}}_{p_r^I}, \tag{1}$$

where: d_0 *is the distance of the desired transmitter;* d_i, $1 \leqslant i \leqslant k$ *are the distances of the* k *interferers; and* P_i *are their transmit power levels.*

Proof: Let Q_0 denote the received power from the desired source and Q_i, $i = 1, \ldots, k$, the received power from k interferers. All the received powers are exponentially distributed, *i.e.*, $p_{Q_i}(q_i) = 1/\bar{Q}_i \, e^{-r_i/\bar{Q}_i}$, where $\bar{Q}_i$ denotes the average received power $\bar{Q}_i = P_i d_i^{-\alpha}$. Conditioned on the total interference $I = \sum i = 1^k Q_i$, the probability of correct reception is a simple exponential, so we have

$$
\begin{aligned}
p_r =& \mathbb{E}_I\big[\mathbb{P}[Q_0 \geqslant \Theta(I+N_0) \mid I]\big] \\
=& \mathbb{E}_I\Big[\exp\Big(-\frac{\Theta(I+N_0)}{\bar{Q}_0}\Big)\Big] \\
=& \int_0^\infty \cdots \int_0^\infty \exp\Big(-\frac{\Theta(\sum_{i=1}^k q_i + N_0)}{\bar{Q}_0}\Big) \prod_{i=1}^k p_{Q_i}(q_i)\, \mathrm{d}q_1 \cdots \mathrm{d}q_k\,,
\end{aligned}
$$

from which (1) follows. ■

Note that the SNR γ_N, the SIR γ_I, and the SINR γ are related as $\gamma = \gamma_N\gamma_I/(\gamma_N+\gamma_I)$, *i.e.*, the SINR is half the harmonic mean of the SNR and SIR, or SINR^{-1}=SNR^{-1}+SIR^{-1}.

This allows an independent analysis of the effect caused by noise and the effect caused by interference. The focus of this paper is put on the noise, *i.e.*, on the first factor in (1). If the load is light (low interference and small collision probability), then SIR$\gg$SNR, and the noise analysis alone provides accurate results. For high load, a separate interference analysis has to be carried out [20–22][1].

In a zero-interference network, the reception probability over a link of distance d at a transmit power P_0, is given by

$$p_r := \mathbb{P}[\gamma_N \geqslant \Theta] = e^{-\frac{\Theta N_0}{P_0\, d^{-\alpha}}}\,. \tag{2}$$

Solving for P_0, the necessary transmit power to achieve a link reliability (or reception probability) p_r is $P_0 = -d^\alpha \Theta N_0/\ln p_r$. It is assumed that the packet transmission time is one time unit, so the energy required for the transmission of one packet over a distance d at reliability p_r is equal to P_0. Thus for the Rayleigh fading channel, we define E_0 to be the energy for a transmission over distance 1 with reliability p_r, *i.e.*, $E_0 = -\Theta N_0/\ln p_r$. Again, in general, $E = d^\alpha E_0$.

[1] Note that *power scaling, i.e.,* scaling the transmit powers of all the nodes by the same factor, does not change the SIR, but (slightly) increases the SINR.

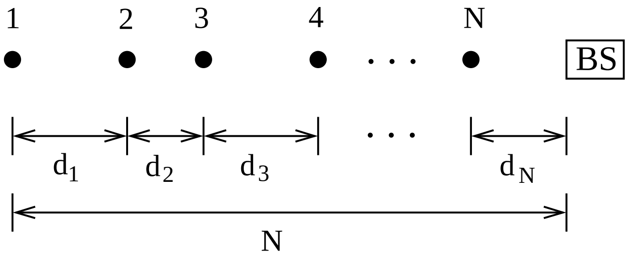


Figure 2: A one-dimensional chain of wireless sensor nodes. It is assumed that the distances sum up to N, *i.e.*, the average distance is 1.

3 Energy-Balancing Strategies

We assume that every sensor node generates an equal amount of traffic of a certain arrival rate that is relayed to the base station along the shortest route. Since optimum routes approximately follow a straight line, the analyses of the strategies proposed in this section can be restricted to one-dimensional chains of N nodes, as shown in Fig. 2. Without loss of generality, it is assumed that the internode distances sum up to N, *i.e.*, the distance between node 1 and the BS is N and the average distance is 1. The strategies are compared with the simple scheme that has equal node distances $d = 1$, equal link reception probabilities p_r, and employs nearest-neighbor routing (node i transmits to node $i+1$ for $i = 1, \ldots, N-1$, and node N transmit to the BS). To compare the total energy consumption, it is sufficient to calculate the energy requirements to forward one packet from each of the nodes to the BS. For the simple strategy, the total energy consumption is

$$E_{\text{tot}} = (1 + 2 + \ldots + N)E_0 = \frac{N(N+1)}{2}E_0 \tag{3}$$

since node i has to forward i packets. For the lifetime, the critical node's energy consumption has to be determined. For the simple strategy, the critical node is node N, and $E_{\max} = E_N = NE_0$.

Although our focus in on the energy consumption, we also consider delay aspects, since delay and transmission rates are closely intertwined with energy. Assuming that spatial reuse of the communication channel permits every q-th link to be active simultaneously, the total number of timeslots to deliver one packet for each node to the BS is given by

$$D = qN - \frac{q(q-1)}{2}\,, \quad \text{for } N \geqslant q-1\,. \tag{4}$$

For $N \leqslant q$, no spatial reuse is possible and the delay corresponds to the total number of transmissions $N(N+1)/2$. For larger N, thanks to the spatial reuse, the delay grows only linearly in N although the number of transmissions increases with N^2.

3.1 Distance variation without rate adjustment

We assume nearest-neighbor routing. The idea is, given a desired rate R (AWGN channel) or link reliability p_r (Rayleigh fading channel), to ensure energy-balancing by adjusting the distances d_i between the nodes. If every node generates one packet, node i has to forward a total of i packets using a total energy of $E_i = i d_i^{\alpha} E_0$. The goal $E_1 = E_2 = \ldots = E_N$ requires that all the factors $i d_i^{\alpha}$ be identical, and the sum of all the distances is N. We find $d_i = d_1 i^{-1/\alpha}$ and

$$d_i = \frac{N}{\sum_{i=1}^{N} i^{-1/\alpha}} \cdot i^{-1/\alpha}\,. \tag{5}$$

To obtain more insight, we derive an accurate approximation based on the harmonic mean-geometric mean inequality. Observe that $d_i = \mathcal{H}(i^{1/\alpha}) \cdot i^{-1/\alpha}$, where $\mathcal{H}(a_i) = N\big(\sum_{i=1}^{N} 1/a_i\big)^{-1}$ is the harmonic mean. d_i can be upperbounded by $\mathcal{G}(i^{-1/\alpha}) i^{-1/\alpha}$, where $\mathcal{G}(a_i) = \Big(\prod_{i=1}^{N} a_i\Big)^{1/N}$ is the geometric mean:

$$d_1 = \mathcal{H}(i^{1/\alpha}) \lessapprox \mathcal{G}(i^{1/\alpha}) = (N!)^{\frac{1}{\alpha N}}\,. \tag{6}$$

With Stirling's approximation $N! \gtrapprox N^N e^{-N} \sqrt{2\pi N}$ (which is within 1% for $N > 8$), this simplifies to

$$d_1 \lessapprox \left(\frac{N}{e}\right)^{\frac{1}{\alpha}} (2\pi N)^{\frac{1}{2\alpha N}}\,. \tag{7}$$

The geometric mean is larger than the actual value (harmonic mean), but Stirling's approximation lower bounds the factorial. In total, (7) is a tight upper bound, as illustrated in Fig. 3. For $\alpha = 4$,

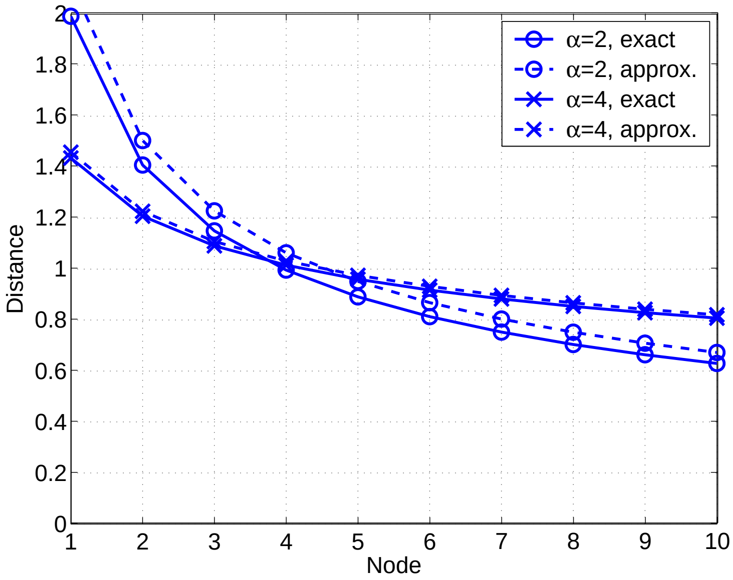


Figure 3: Internode distances for distance variation strategy without rate adjustment. N=10 nodes, $\alpha = 2, 4$.

its accuracy is within 3% for $N \leqslant 100$, for $\alpha = 3$, it is within 5% for $N \leqslant 60$, and for $\alpha = 2$ within 10% for $N \leqslant 30$. Dropping the second factor in (7) yields the even simpler expression $d_1 \approx (N/e)^{1/\alpha}$, which turns out to be a tight lower bound for $N \lessapprox 7\alpha$.

The energy consumption at each node is $d_1^\alpha E_0 \approx (N/e)E_0$. So, the gain (reduction) in total energy consumption is

$$G_{\text{tot}} \approx 1 - \frac{N^2/e}{N(N+1)/2} = \frac{2N}{e(N+1)} \approx 1 - 2/e \approx 26\% \,. \tag{8}$$

The gain in lifetime is considerably higher, namely a factor of e, since node N now only consumes N/e energy units rather than N.

This strategy leads to a non-uniform distribution of the sensor nodes, which may not be desirable. However, the distribution is not too far from uniform, in particular for high path loss exponents, as is manifested by the small variance of d_i. For $N = 10$ and $\alpha = 2$, the variance is about 0.18. For $\alpha = 3, 4, 5$, the variances are 0.073, 0.039, 0.024.

3.2 Distance variation with rate adjustment

In this approach, the rate of transmission is adjusted based on the amount of data to be forwarded. If at every hop the full amount of data is to be transmitted in a single timeslot, the rate needs to increase linearly with i, *i.e.*, $R_i \propto i$. Since the transmit power should be the same for all nodes, the SNR $\gamma_i = c d_i^{-\alpha}$ can only be increased by decreasing the internode distance. From $R_i = \log_2(1 + c d_i^{-\alpha}) = iR_1$ and $c = d_1^\alpha \gamma_1$ follows

$$d_i^\alpha = d_1^\alpha \frac{\gamma_1}{\gamma_i} = d_1^\alpha \frac{2^{R_1} - 1}{2^{iR_1} - 1} \,. \tag{9}$$

The distances d_i sum up to N, so $d_1 = N / \sum_{i=1}^N (\gamma_1/\gamma_i)^{1/\alpha}$ and

$$d_i = \frac{N \gamma_i^{-1/\alpha}}{\sum_{i=1}^N \gamma_i^{-1/\alpha}} = \frac{N(2^{iR_1} - 1)^{-1/\alpha}}{\sum_{i=1}^N (2^{iR_1} - 1)^{-1/\alpha}} \,. \tag{10}$$

The rate R_1 of the first link is a design parameter. In the following, approximations for d_i and the energy consumption are derived, discussing the high rate ($R_1 \gtrapprox 1$) and the low rate ($R_1 \approx 1/N$) cases separately.

High rate: Since $2^{iR_1} \gg 1$ for most i, the sum in the denominator of (10) can be approximated as a geometric sum, *i.e.*, $\sum_{i=1}^N 2^{-iR_1/\alpha} = (1 - 2^{-R_1 N/\alpha})/(2^{R_1/\alpha} - 1)$, which yields

$$d_i \approx \frac{N}{(2^{iR_1} - 1)^{1/\alpha}} \cdot \frac{2^{R_1/\alpha} - 1}{1 - 2^{-R_1 N/\alpha}} \,. \tag{11}$$

The term $2^{-R_1 N/\alpha}$ is negligible for large N/α, which shows that the distances increase approximately linearly with N and that d_N goes to zero quickly as N grows. For $R_1 = 1$ (which implies $E_0 = N_0$), we obtain $d_1 \approx N(2^{1/\alpha} - 1)$ and $d_N \approx N 2^{-N/\alpha}(2^{1/\alpha} - 1)$. This case is illustrated in Fig. 4. The energy consumption at each node is $E = E_1 = d_1^\alpha N_0 = N^\alpha (2^{1/\alpha} - 1)^\alpha N_0$, so the total energy grows with $N^{\alpha+1}$, and the total delay is N.

Low rate: Since $2^{iR_1} - 1 \ll 1$ for most i, we employ the first order Taylor approximation $2^{iR_1} - 1 \approx iR_1 \ln 2$ and the harmonic mean-geometric mean inequality. The geometric mean $\mathcal{G}((2^{iR_1} -$

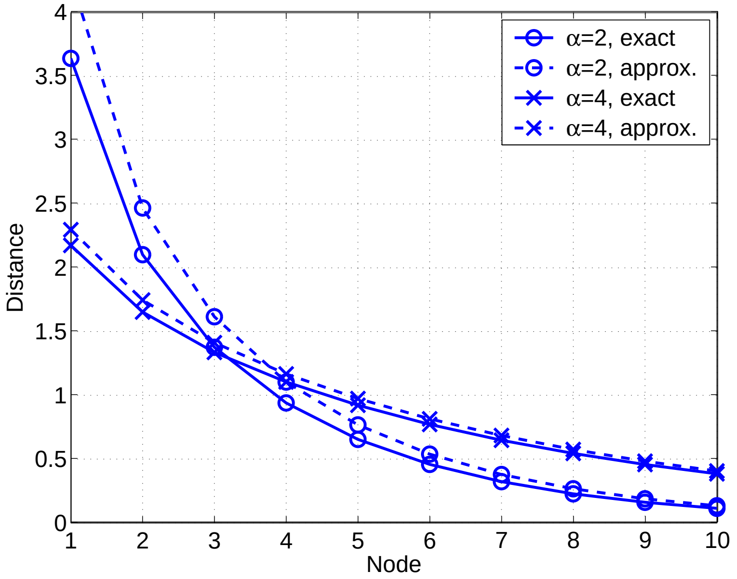


Figure 4: Internode distances for distance variation with rate adjustment for $R_1 = 1$ (high rate), N=10 nodes, $\alpha = 2, 4$.

$1)^{1/\alpha})$ is $(R_1 \ln 2)^{1/\alpha}(N!)^{1/(\alpha N)}$. Applying a crude version Stirling's formula, $N! \approx (N/e)^N$, and noting that $\ln 2/e \approx 1/4$, we obtain the simple approximation $\mathcal{G} \approx (R_1 N/4)^{1/\alpha}$. Therefore

$$d_i \approx \left(\frac{R_1 N}{4(2^{iR_1} - 1)}\right)^{\frac{1}{\alpha}} (2\pi N)^{\frac{1}{2\alpha N}}\,, \tag{12}$$

which is shown in Fig. 5 together with the exact value from (10). The energy consumption at each hop is

$$E_i = E = d_i^\alpha E_0 \approx \frac{R_1 N}{4} N_0\,. \tag{13}$$

For $R_1 = 1/N$, this simplifies to $E \approx N_0/4$, so the total energy consumption grows only linearly with N. On the other hand, the delay is N^2 for this low-rate scheme.

Remarks:

- The low-rate case looks quite similar to the case without rate adaptation. The variance is just slightly larger in the low-rate case, namely 25%. In the high-rate case, the variance may be unacceptably high, hence it would be difficult to achieve good coverage.

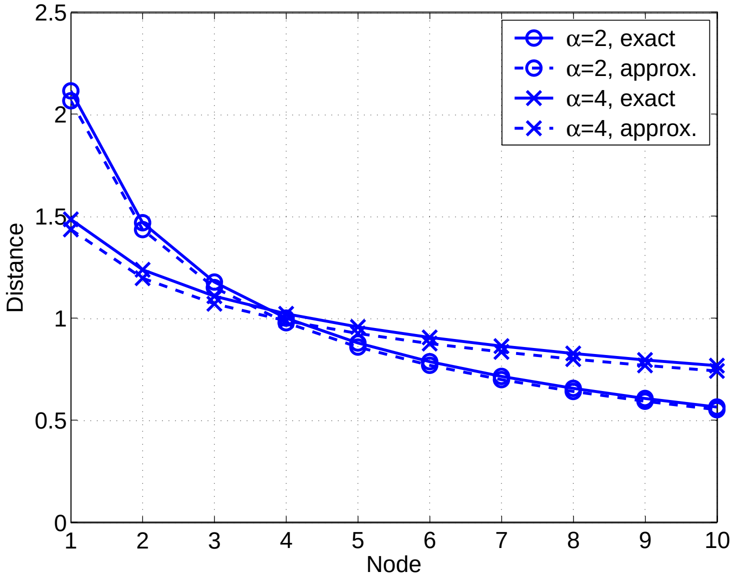


Figure 5: Internode distances for distance variation with rate adjustment for $R_1 = 1/N$ (low rate), N=10 nodes, $\alpha = 2, 4$.

- Between no rate adaptation and full rate adaptation, there is a host of other possibilities. Depending on the desired performance on the energy/delay curve, the rates of the nodes can be adapted.

3.3 Balanced data compression

If the internode distances are all equal ($d_i = 1$), the incoming data flows may be compressed to ensure that every node in the path has to transmit the same number of packets. The justification for this approach is the correlation between the sensor readings of neighboring nodes. Hence, data fusion may be applied to reduce the amount of data to be transmitted. The goal is to ensure that every packet experiences the same compression factor, irrespective of its origin. At each node i, the incoming data is compressed by a factor of a_i, while the locally generated data is compressed by $b_i = 1 - a_i$. Equal compression is achieved when $b_i = 1/i$, since the total compression factor for a packet generated at node i is

$$\beta_i = b_i \cdot \prod_{k=i+1}^{N} a_i = \frac{1}{i} \cdot \prod_{k=i+1}^{N} \left(1 - \frac{1}{k}\right) = \frac{1}{N} \;\; \forall i\,. \tag{14}$$

This way, every node transmits only one packet, so the total energy consumption is NE_0, and the gain in lifetime is a factor of N. On the other hand, the accuracy of the received data will suffer. The total transmission delay is N, but the computational or processing delay also needs to be taken into account.

3.4 Routing

We again assume equal distances $d_i = 1$ between the nodes but no longer restrict the network to strict nearest-neighbor routing. Instead, we assume that node i transmits the locally generated traffic to the next neighbor with probability a_i and directly to the sink with probability $b_i = 1 - a_i$. Relay traffic will always be forwarded to the next node. The goal is to choose a_i to achieve energy balancing.

All energies in the following derivation are normalized by E_0. The energy consumption at node i is

$$\begin{aligned} E_i =& (N - i + 1)^\alpha b_i + \sum_{k=1}^{i} a_k & (15) \\ =& i + \left((N - i + 1)^\alpha - 1\right) b_i - \sum_{k=1}^{i-1} b_k\,. & (16) \end{aligned}$$

$b_N = 0$, as node N always transmits directly to the BS. $E_N = N - b_1 - b_2 - \ldots - b_{N-1} = a_1 + a_2 + \ldots + a_N$. The $N - 1$ unknowns can thus be determined by solving

$$\begin{bmatrix} N^\alpha & 1 & \ldots & 1 \\ 0 & (N-1)^\alpha & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 2^\alpha \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N-1} \end{bmatrix} = \begin{bmatrix} N-1 \\ N-2 \\ \vdots \\ 1 \end{bmatrix} \tag{17}$$

to equalize the energy consumption at every node to E_N. For a network with 5 nodes, the values for $b_1, \ldots, b_5$ are 0.0301, 0.0438, 0.0694, 0.1250, 0 for $\alpha = 3$. Since some packets are routed to the BS in a single hop, the total energy consumption is bigger than in the simple strategy. For $N = 10$, the additional energy consumption is between 60% ($\alpha = 2$) and 80% for ($\alpha = 5$). On the other hand, there is a slight gain in network lifetime, as the direct routing of some of the packets reduces the burden on node N. Therefore the sum of the a_i's is smaller than N. For 10 nodes, the increase in lifetime is 0.5% for $\alpha = 5$ and 14% for $\alpha = 2$. For $N \to \infty$, the total additional energy consumption reaches 100% and the gain in lifetime vanishes[2]. Hence this strategy is useful for smaller N, or if there are a some high-priority packets that have to be delivered with minimum delay. The average delay for packets generated at node i is $a_i(N-i+1)+(1-a_i) = a_i(N-i)+1$.

3.5 Equalization of the end-to-end reliability

While the last to strategies are applicable to both AWGN and fading channels, here we focus only on Rayleigh fading channels. So far, we ignored the fact that the end-to-end reliability of a multihop path is the product of the reception probabilities of the links. With constant link reception probabilities p_r, a packet traveling over k hops only arrives at the sink with a probability p_r^k.

We investigate a strategy where every packet, irrespective of how far away from the base station it is generated, has the same end-to-end probability p_{EE} to arrive at the base station. This *equal-end-to-end-probability* or *reliability balancing* strategy is henceforth referred to as strategy A, whereas the simple *equal-power* strategy is denoted as strategy B.

Analysis of strategy A. If the desired end-to-end reliability is p_{EE}, the link probability in

[2] It is easily established that $b_k \leqslant (N-k)\cdot(N-k+1)^{-\alpha}$. With $E_N > \sum_{k=1}^{N-1}(1-b_k)$, we get $E_N > \sum_{k=1}^{N-1} 1 - k\cdot(k+1)^{-\alpha}$. For $\alpha \geqslant 2$, the terms in the sum approach 1 for large k, thus $E_N \to N$ for large N, which is the same as in the simple strategy.

a $k-$hop connection is $p_{L_k} = p_{\mathrm{EE}}^{1/k}$. Accordingly, the transmit power at each hop in a $k-$hop connection with equal distances $d_i = 1$ is[3]

$$P_k^{\mathsf{A}} = \frac{\Theta N_0}{-\ln(p_{\mathrm{EE}}^{1/k})} = k \cdot \frac{\Theta N_0}{-\ln p_{\mathrm{EE}}} \,. \tag{18}$$

Clearly, a transmission over k hops requires a k times higher transmit power level (at each hop) than a transmission over one hop with the same probability. Thus the total energy needed for a packet to travel from node $N-k+1$ to the base station is proportional to k^2. Note that a single large hop of length k would require an energy proportional to k^α. Let E_0^{A} denote the energy required to transmit one packet over one hop of distance 1 with a reliability of p_{EE}, *i.e.*, $E_0^{\mathsf{A}} := \Theta N_0/(-\ln p_{\mathrm{EE}})$. Using nearest-neighbor routing, the energy consumption is:

Node	Energy consumption		(in units of E_0^{A})
1	N	$=$	N
2	$N + (N-1)$	$=$	$2N - 1$
3	$(2N-1) + (N-2)$	$=$	$3N - 3$
4			$4N - 6$
$\vdots$	$\vdots$		$\vdots$
i			$iN - \frac{i(i-1)}{2}$

The total energy consumption (assuming one packet is generated at every node) is

$$\begin{aligned}
\frac{E_{\mathrm{tot}}^{\mathsf{A}}}{E_0^{\mathsf{A}}} &= \sum_{i=1}^{N} \Big(iN - \frac{i(i-1)}{2} \Big) \\
&= N \cdot \frac{N(N+1)}{2} - \underbrace{\Big(0 + 1 + 3 + 6 + \ldots + \frac{N(N-1)}{2} \Big)}_{S} \\
&= N \cdot \frac{N(N+1)}{2} - \Big(\frac{N(N-1)}{2} + \frac{N(N-1)(N-2)}{6} \Big) \\
&= \frac{N^3}{3} + \frac{N^2}{2} + \frac{N}{6} \,,
\end{aligned} \tag{19}$$

[3] Note that this implies that a single node uses different power levels that depend on the origin of a packet. The power levels are assigned to flows, not to nodes.

where we have exploited the fact that the sum S in the second line is an arithmetic series of order 2 with $q_0 = 0$, $\Delta q_0 = \Delta^2 q_0 = 1$.

Analysis of strategy B. Here, all nodes transmit at a fixed power level, corresponding to a fixed link reception probability p_r. For a fair comparison, it is assumed that the application dictates a *minimum* end-to-end reliability p_{EE}. Packets generated at node i have to travel over $k := N - i + 1$ hops, resulting in an end-to-end reliability of p_r^k. To ensure that the packets from the farthest node (node 1) arrive with probability p_{EE}, a link reception probability of $p_r = p_{\mathrm{EE}}^{1/N}$ is required. The energy per hop is in this case is (cf. (3))

$$E_0^{\mathsf{B}} = d^\alpha \Theta N_0 / (-\ln P_L) = N \cdot d^\alpha \Theta N_0 / (-\ln p_{\mathrm{EE}}) = N E_0^{\mathsf{A}} .$$

The total energy consumption of the network (assuming one packet is generated at every node) is

$$E_{\mathrm{tot}}^{\mathsf{B}} = \frac{N(N+1)}{2} E_0^{\mathsf{B}} = \Big(\frac{N^3}{2} + \frac{N^2}{2}\Big) E_0^{\mathsf{A}} . \tag{20}$$

Comparison. The ratio between the energy consumption of the two strategies is

$$\frac{E_{\mathrm{tot}}^{\mathsf{A}}}{E_{\mathrm{tot}}^{\mathsf{B}}} = \frac{\frac{N^2}{3} + \frac{N}{2} + \frac{1}{6}}{\frac{N^2}{2} + \frac{N}{2}} = \frac{2}{3} + \frac{1}{3N} . \tag{21}$$

For large N, this ratio approximates 2/3, hence the gain in total energy consumption for the reliability balancing strategy is 33%. More important and more significant is the gain in network lifetime, which is determined by the lifetime of the critical node N. In strategy A, the energy consumption at node N is $E_N^{\mathsf{A}} = E_0^{\mathsf{A}}(N^2 + N)/2$, whereas for B, it is $E_N^{\mathsf{B}} = E_0^{\mathsf{B}} N = E_0^{\mathsf{A}} N^2$. The ratio is $2N/(N+1)$, thus the gain in network lifetime approaches 2 for large N. Figure 6 compares the two strategies. For strategy B, the energy consumption increases linearly with the node number, and the end-to-end probability for a packet generated at node i increases monotonically. For scheme A, the energy consumption is more balanced, and the end-to-end probability is constant.

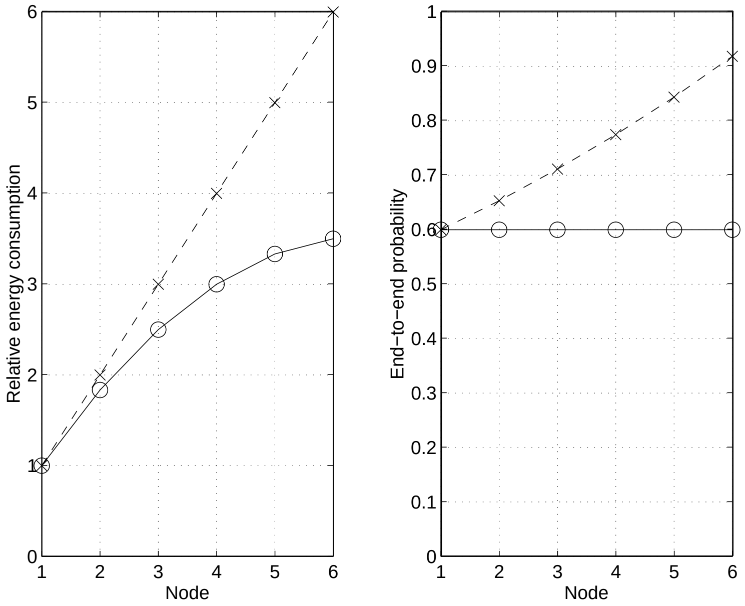


Figure 6: Comparison of the equal-power strategy B (dashed) and the equal-end-to-end-probability strategy A (solid) for $p_{\mathrm{EE}} = 0.6$. The left plot shows the relative energy consumption when every node generates one packet, the right plot the end-to-end probabilities for traffic generated at nodes $1, 2, \ldots, 6$.

4 Concluding Remarks

For sensor networks, where the destination of all the information gathered at the sensor nodes is a single base station, the traffic pattern is highly non-uniform, since the nodes close to the base station have to relay all the data packets. Consequently, those nodes are the first to run out of battery, thereby restricting the lifetime of the network. Several strategies have been proposed to balance the energy consumption, each of them having their application-dependent strengths and weaknesses. The strategies are not mutually exclusive, hence two or more may be combined into hybrid schemes. For example, since the distance variation strategy does not provide reliability balancing, both could be applied jointly to achieve both energy and reliability balancing.

The analyses are based on a AWGN channel model and on a Rayleigh fading model. Under Rayleigh fading, noise issues and interference issues can be analyzed separately, and that the reception probability is an exponential function of the transmit power. This non-zero probability of a packet loss even if nodes are close (or within "transmission range") is often ignored. It entails that a transmission over k hops requires a k times higher transmit power at each hop to guarantee the same end-to-end probability as a single-hop transmission. An important consequence is that the energy benefit of multihop routing becomes less significant. Indeed, since the energy consumption is proportional to k^2 in the hop-by-hop transmission and proportional to k^α in a single-hop scheme (see Section 3.5), the benefit vanishes for $\alpha = 2$.

Acknowledgments

The partial support of the DARPA/IXO-NEST Program (AF-F30602-01-2-0526) and NSF (ECS02-25265 and ECS03-29766) is gratefully acknowledged.

References

[1] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 551–558, 2000.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, Mar. 2002.

[3] A. Ephremides, "Energy Concerns in Wireless Networks," *IEEE Wireless Communications*, vol. 9, pp. 48–59, Aug. 2002.

[4] "Mesh networks." `http://www.meshnetworks.com`.

[5] J. Jun and M. L. Sichitiu, "The Nominal Capacity of Wireless Mesh Networks," *IEEE Wireless Communications*, vol. 10, pp. 8–14, Oct. 2003.

[6] Y.-D. Lin and Y.-C. Hsu, "Multihop Cellular: A New Architecture for Wireless Communications," in *IEEE INFOCOM*, vol. 3, pp. 1273–1282, 2000.

[7] K. M. Pepe and B. R. Vojcic, "Cellular Multihop Networks and the Impact of Routing on the SNIR and Total Power Consumption," in *Workshop on Multiaccess, Mobility and Teletraffic for Wireless Communications conference (MMT'02)*, (Rennes, France), June 2002.

[8] V. Sreng, H. Yanikomeroglu, and D. Falconer, "Coverage Enhancement through Two-Hop Relaying in Cellular Radio Systems," in *Wireless Communications and Networking Conference (WCNC)*, vol. 2, pp. 881–885, 2002.

[9] V. Sreng, H. Yanikomeroglu, and D. Falconer, "Relayer Selection Strategies in Peer-to-Peer Relaying in Cellular Radio Networks," in *IEEE Vehicular Technology Conference (VTC'03 Fall)*, (Orlando, FL), Oct. 2003.

[10] H. Li, M. Lott, M. Weckerle, W. Zirwas, and E. Schulz, "Multihop Communications in Future Mobile Radio Networks," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept. 2002.

[11] J. A. Silvester and L. Kleinrock, "On the Capacity of Multihop Slotted ALOHA Networks with Regular Structure," *IEEE Transactions on Communications*, vol. COM-31, pp. 974–982, Aug. 1983.

[12] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Transactions on Communications*, vol. COM-32, pp. 246–257, Mar. 1984.

[13] J. L. Wang and J. A. Silvester, "Maximum Number of Independent Paths and Radio Connectivity," *IEEE Transactions on Communications*, vol. 41, pp. 1482–1493, Oct. 1993.

[14] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, Mar. 2000.

[15] A. J. Goldsmith and S. B. Wicker, "Design Challenges for Energy-Constrained Ad Hoc Wireless Networks," *IEEE Wireless Communications*, vol. 9, pp. 8–27, Aug. 2002.

[16] E. S. Sousa and J. A. Silvester, "Optimum Transmission Ranges in a Direct-Sequence Spread-Spectrum Multihop Packet Radio Network," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 762–771, June 1990.

[17] D. A. Maltz, J. Broch, and D. B. Johnson, "Lessons from a Full-Scale Multihop Wireless Ad Hoc Network Testbed," *IEEE Personal Communications*, vol. 8, pp. 8–15, Feb. 2001.

[18] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An Empirical Study of Epidemic Algorithms in Large Scale Multihop Wireless Networks,"

2002. Intel Research Report IRB-TR-02-003. Available at `http://www.intel-research.net/Publications/Berkeley/050220021703_19.pdf`.

[19] T. S. Rappaport, *Wireless Communications – Principles and Practice.* Prentice Hall, 1996. ISBN 0-13-375536-3.

[20] M. Haenggi, "Probabilistic Analysis of a Simple MAC Scheme for Ad Hoc Wireless Networks," in *IEEE CAS Workshop on Wireless Communications and Networking*, (Pasadena, CA), Sept. 2002.

[21] X. Liu and M. Haenggi, "Throughput Bounds and Energy Consumption of Mobile Multihop Networks," in *IEEE Vehicular Technology Conference (VTC'04 Fall)*, (Los Angeles, CA), Sept. 2004.

[22] J. Venkataraman and M. Haenggi, "Optimizing the Throughput in Random Wireless Ad Hoc Networks," in *42st Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, IL), Oct. 2004.

Twelve Reasons not to Route over Many Short Hops

Martin Haenggi
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
E-mail: mhaenggi@nd.edu

Abstract— **For multihop wireless networks, a fundamental question is whether it is advantageous to route over many short hops (short-hop routing) or over a smaller number of longer hops (long-hop routing). Short-hop routing gained a lot of support, and its proponents mainly produce two arguments: reduced energy consumption and less interference. Both arguments stem from an oversimplified analysis that is based on inaccurate channel models and neglects delay, end-to-end relability, bias power consumption, the impact of channel coding, mobility, and routing overhead. In this paper, we shed more light on these issues by listing twelve reasons why short-hop routing is not as beneficial as it seems to be. The conclusion is that for many networks, long-hop routing is in every aspect a very competitive strategy.**

I. Introduction

For certain wireless networks, such as ad hoc and sensor networks, a fundamental question is whether it is advantageous to route over many short hops (*short-hop routing* or, in the extreme case, nearest-neighbor routing) or over a smaller number of longer hops (*long-hop routing*). Recently, this debate extended to multihop extensions of WLANs [1] and multihop cellular networks [2]. Short-hop routing gained a lot of support, and its proponents mainly produce the following two arguments:

1. Energy consumption. If a long hop of distance d is divided into n hops of distance d/n, the energy benefit is often assumed to be $n^{\alpha-1}$, where α is the path loss exponent.
2. Capacity. The shorter the hops, the higher the transport capacity in an interference-limited network [3].

The first argument stems from an oversimplified analysis of the energy consumption and neglects important issues such as delay, end-to-end reliability, and bias power consumption. The second argument is only valid as long as the connectivity of the network is guaranteed; it was derived for an increasingly dense network that takes advantage of the singularity of the attenuation $d^{-\alpha}$ at $d = 0$, which may lead to the unrealistic situation that the received power exceeds the transmitter power; and it neglects delay, too. In this paper, we shed more light on these issues by listing twelve reasons why short-hop routing is not as beneficial as it seems to be. Some of the reasons have been mentioned in other work, but this is, to the best of our knowledge, the first comprehensive collection.

Often, a *disk model*[1] is used for the analysis of wireless networks, where a transmission is either 100% successful or fails completely, depending on whether the distance is smaller or larger than the so-called transmission radius. More realistic is the *threshold model*[2], where a certain signal-to-noise-and-interference ratio (SINR) is needed for successful transmission. Still, for AWGN channels, the threshold model yields 0% or 100% probability and should therefore be used with great care. To get accurate results, reception probabilities should be based on bit, block, and packet errors rates, taking into account the error correction capabilities of the channel code.

We will demonstrate that by discarding the disk model and directly focusing on SINR levels, many advantages of long-hop routing become apparent.

II. Network and Link Model

Part of our discussion applies to many types and classes of networks and wireless channels. However, to be concrete, we often focus on networks with random node distribution and Rayleigh fading channels.

A. Node Distribution and Generic Routing

The analytical results are derived for networks whose nodes constitute a Poisson point process in the plane. Note that for infinite networks, the Poisson point process corresponds to a uniform distribution [4], [5], and for large networks, the two distributions are equivalent for all practical purposes.

Many different routing algorithms exist for ad hoc networks [6], [7], but common to all of them is the fact that at each hop, progress shall be made towards the destination. This generic routing strategy is illustrated in Fig. 1. If the nearest neighbor within a certain sector of the source-destination axis is chosen as the next relay, this is certainly an instance of *short-hop* routing. If many nearby neighbors are skipped and a node

[1] Also called protocol model [3].
[2] Also denoted as physical model [3].

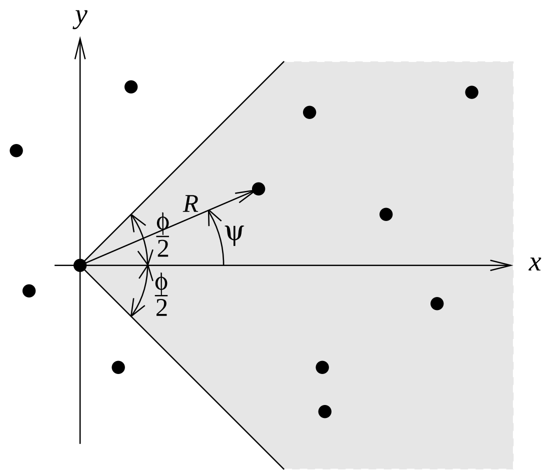


Fig. 1. Part of a Rayleigh network with the source at the origin and the x-axis pointing towards the destination node. R denotes the distance to the nearest neighbor with in a sector ϕ around x, and ψ is its argument. Hence (R, ψ) are the polar coordinates of the nearest neighbor within a sector ϕ, and (X, Y) are its Cartesian coordinates.

transmits directly to a more distant neighbor, we speak of *long-hop* routing.

Note that the distance to the n-th nearest neighbor that lies within the desired sector can easily be determined from the Poisson assumption. The probability density of the distance to the n-th neighbor in a sector ϕ is [8]

$$p_{R_n}(r) = r^{2n-1}\left(\frac{\phi}{2}\right)^n \frac{2}{(n-1)!}\, e^{-r^2\phi/2} \qquad (1)$$

Since p_{R_n} is a Rayleigh distribution for $n = 1$, it can be considered a generalized Rayleigh distribution. Similarly, in one dimension, the Erlang distribution is a generalized exponential distribution. So, the transition from one dimension to two dimensions simply entails a multiplication by $r\phi$ (that comes from the inner derivative of the exponential part) in the distributions of the node distances.

B. Link Model

Although some of the reasons listed in the next section also pertain to the disk model, we will mostly use a Rayleigh block fading channel model. Assuming that a certain SINR level Θ is required for successful reception (threshold model), the reception probability of successful packet reception can be expressed as [8]

$$p_r = \underbrace{\exp\Big(-\frac{\Theta N}{P_0 d_0^{-\alpha}}\Big)}_{p_r^N} \cdot \underbrace{\prod_{i=1}^{k} \frac{1}{1+\Theta\frac{P_i}{P_0}\big(\frac{d_0}{d_i}\big)^{\alpha}}}_{p_r^I}\,, \qquad (2)$$

where N is the noise power, P_0 the transmit power of the desired transmitter, P_i, $i > 0$ the transmit powers of the k interferers (who transmit in the same time slot), d_0 the distance from the receiver to the desired transmitter, and d_i, $i > 0$, the distance to the interfering transmitters. α is the large-scale path loss exponent. In the expression (2), we can identify two parts: p_r^N, which is the reception probability in an interference-free network since it only depends on the noise, and p_r^I, which is the reception probability in a noise-free network since it only depends on the interference. This property of the Rayleigh fading channel enables the separate analysis of noise and interference effects.

III. The Twelve Reasons

Clearly, the most compelling reason against short-hop routing is the end-to-end delay. However, we do not consider delay in itself an argument for long-hop routing, since energy and delay can be traded off against each other. So, for a fair comparison, both schemes shall satisfy the same energy and delay requirements. We may fix the energy consumption and search for the protocol with the smallest delay, or we may impose a delay constraint and determine which protocol consumes the least amount of energy.

A. Interference

According to [9], "It is unclear whether more interference is caused by a single transmission at higher power or multiple transmissions at lower power". Indeed, a shorter transmission at higher power may permit more efficient reuse of the communication channel. If the total radiated energy (product of power and duration) of is a good indicator for interference, this boils down to an energy consumption problem. However, it must not be forgotten that *the SIR does not depend on absolute power levels*. If all nodes scale their power by the same factor $q > 1$, all the SIR levels remain constant, but the SINR levels will increase, which can be easily seen from (2). So, increasing all transmit power levels does not have a negative impact on any packet reception probability in the network (on the contrary), in stark contrast to what is predicted by the disk model. This indicates that *long-hop transmission does not inherently cause more interference*.

B. End-to-end Reliability

Under the disk model, reception probabilities are either 100% or 0%. If every receiver is in its desired transmitter's disk, the end-to-end reliability is always 100%, which is clearly not realistic, since packet errors or bit errors accumulate. In the Rayleigh block fading case, the end-to-end reception probability p_n over an n-hop route follows from (2):

$$p_n = \prod_{i=1}^{n} e^{-\Theta/\bar{\gamma}_i} = e^{-\Theta\sum_{i=1}^{n}\frac{1}{\bar{\gamma}_i}} \qquad (3)$$

where $\bar{\gamma}_i$ denotes the mean SNR at link i. So, to achieve a desired end-to-end reliability with short-hop routing, the relay nodes need to transmit at a higher power. This compensates, at least partially, for the loss in SNR.

Consider the case of a one-dimensional chain of equidistant nodes with distance d (this is the optimum scenario for short-hop routing). Let E_0 be the energy required for a transmission over distance d with probability P_{EE}, *i.e.*, $E_0 := -d^\alpha \Theta N / \ln P_{EE}$. Covering the total distance in one single hop requires an energy of $E_1 = n^\alpha E_0$. In the multihop case with n hops, a reception probability $p_r = \sqrt[n]{P_{EE}}$ is required at each hop. Since $\ln P_{EE} = n \ln p_r$, the total energy in this case is $E_n = n \cdot n\, E_0$. So, for $\alpha = 2$, there is no benefit in multihop routing.

If the same end-to-end delay is permitted, a number of transmission (at lower power) are possible in the long-hop case. For the block fading channel, this results in a time diversity benefit, and the energy advantage of short-hop routing vanishes also for higher path loss exponents, in particular for high end-to-end reliabilities or when channel state information is available at the transmitter.

C. Capacity and Channel Coding

For optimum coding in AWGN channels, a change from a nominal capacity $C_0 := \log_2(1 + P_0/N)/2$ to C_0/q results in an energy consumption of $E(q) = qP(q) = qN(2^{2C_0/q} - 1)$, which is strictly monotonically decreasing in q.[3] Note that $C_0/q =: R$ is the information theoretic rate (bits/symbol), and that the gain from using longer codes is higher for higher rates.

Assume that a (normalized) distance 1 is covered by an n-hop route for some small n, say $1 \leqslant n \leqslant 5$. In this case, spatial reuse is not possible, so a simple TDMA MAC scheme will perform optimally. Since there is no interference in this case, the analysis is based on AWGN channels. Let R denote the bandwidth-normalized end-to-end rate for the n-hop route. We find

$$R = \frac{1}{n} \log_2 \left(1 + \frac{E_s}{N_0} n^\alpha \right) , \tag{4}$$

where $E_s = E_b R$ is the energy per symbol and N_0 is the noise spectral density. By dividing the distance into hops of length $1/n$, the SNR increases by n^α. On the other hand, the end-to-end rate is reduced by a factor of n. Conversely, the n-hop scheme needs to transmit at a per-hop rate of nR to achieve the same end-to-end rate. The question is which is the optimum n given a certain desired end-to-end rate. This problem is addressed in some detail in [10]. Here we note that since the rate loss in (4) is linear while the gain from the increased SNR is only proportional to $\log_2(n)$, there exists a certain rate R_n at which the bit-energy-to-noise ratio E_b/N_0 is the same for n-hop and $n+1$-hop routing. Above R_n, n-hop routing performs better, and below R_n, $(n+1)$-hop routing is better. Focusing on the case of single-hop and two-hop routing, we find from (4) that $R_1 = \log_2(2^\alpha - 1) < \alpha$. So, as a simple rule of thumb, we can say that whenever the desired rate is higher than the path loss exponent, single-hop routing achieves the highest capacity.

More generally, it can be shown that for $R \to 0$, there exists an asymptotic per-hop spectral efficiency

$$S := \lim_{R \to 0} R\, n_{\text{opt}}(R) \tag{5}$$

that only depends on α. It is given by

$$S = \frac{\mathrm{W}(-\alpha e^{-\alpha}) + \alpha}{\ln 2} \lessapprox \frac{\alpha(1 - e^{-\alpha})}{\ln 2} , \tag{6}$$

where $\mathrm{W}(\cdot)$ denotes the (principal branch of the) Lambert W function and the bound stems from the first order Taylor expansion. The primary use of this asymptotic spectral efficiency is to determine the optimum hop number: It can be shown that with good accuracy

$$n_{\text{opt}}(\alpha, R) \approx [S(\alpha)/R] , \tag{7}$$

where $[x]$ denotes the nearest positive integer to x. Using the upper bound (6), we find

$$n_{\text{opt}}(\alpha, R) \leqslant \left[\frac{\alpha(1 - e^{-\alpha})}{R \ln 2} \right] , \tag{8}$$

which conveniently yields the optimum number of hops for any R and α.

If channel coding is taken into account, multihop is further penalized due to the necessary encoding and decoding at each hop.

D. Total Energy Consumption

It is often assumed that a reduction of the *transmit energy* yields a proportional reduction of the total energy consumption. Even without taking into account *receive* energy, this is not true for any practical power amplifier. In particular in low-power transceivers, the local oscillators and bias circuitry will dominate, so that short-hop routing does not yield any energy benefit if a more distant relay node can be reached with sufficient reliability [11]. For random networks, relatively high peak power levels are necessary to keep the network connected [12], and short-hop routing would require a substantial backoff on the average, resulting in poor power efficiency.

E. Path Efficiency in Random Networks

Routes in random networks cannot follow straight lines. The path efficiency, defined as the ratio of Euclidean distance of the end nodes and the travelled distance, is higher if longer hops are used. Consider the generic routing strategy in Fig. 1. For nearest-neighbor routing in a sector ϕ the expected path efficiency η for a long connection is $\mathbb{E}[\cos \Psi]$, so

$$\eta(\phi) = \mathbb{E}[\cos \Psi] = \frac{2}{\phi} \int_0^{\phi/2} \cos \psi \, d\psi \tag{9}$$

$$= \frac{2}{\phi} \sin \left(\frac{\phi}{2} \right) \approx 1 - \frac{\phi^2}{24} , \tag{10}$$

[3] The increase in transmission length is only linear in q, while the power can be reduced exponentially.

where the approximation is the second-order Taylor expansion. Now assume that this nearest-neighbor path has been established, and that, instead of routing through every node, a long-hop strategy is applied where only every n-th node is used as a relay. With increasing n, the argument Ψ from the origin to the n-th node tends to be Gaussian distributed with variance $V(\phi, n)$. Since the support of the pdf of Ψ is always $[-\phi/2, \phi/2]$,[4] the variance decreases inversely proportional to n, *i.e.*, $V(\phi, n) \approx \frac{\phi^2}{12n}$. So, for large n, we get[5]

$$\eta_n(\phi) = \mathbb{E}[\cos\Psi] \tag{11}$$

$$\approx \frac{\sqrt{6n}}{\phi\sqrt{\pi}} \cdot \int_{-\infty}^{\infty} \cos\psi \, e^{-6n\left(\frac{\psi}{\phi}\right)^2} \mathrm{d}\psi \tag{12}$$

$$= e^{-\frac{\phi^2}{24n}} \tag{13}$$

$$\approx 1 - \frac{\phi^2}{24n}\,, \tag{14}$$

which shows that the path efficiency grows monotonically with n and goes to 1 for large n. The energy penalty caused by deviations from the optimum path is

$$\frac{1}{\eta_n(\phi)^\alpha} \approx 1 + \frac{\alpha\phi^2}{24n}\,. \tag{15}$$

This shows that the energy penalty can be made independent of α, if only every $n = \alpha$-th node is used.

F. Sleep Modi or Cooperation

If neighboring nodes are not used as relays, they can either be put to sleep, or they can assist the transmission by cooperation [13], [14] or retransmission (*e.g.*, if an ACK packet is not received by the source).

G. Routing Overhead and Route Maintenance

In [9], it is pointed out that (when we replace a larger number of short hops by a smaller number of long hops) “It is far from clear what happens to the overall transmission energy, since to implement a nearest-neighbor policy, significantly augmented overhead control traffic will be required to coordinate the establishment of the routing paths and access control protocols across the entire network.”

In a first order approximation, the control traffic for routing and route maintenance is proportional to the number of nodes in the route. Also, the probability of a route break due to energy depletion and node failure clearly increases with the number of nodes involved, as well as the memory requirements for the routing tables.

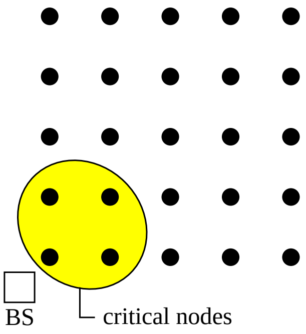


Fig. 2. The *critical area* of a sensor network is determined by the nodes that can reach the base station in a single hop. If it consists of only a few nearest neighbors of the base station, these nodes will severely limit the lifetime of the network since they have to relay the whole traffic.

H. Route Longevity in Mobile Environments

The SNR of short-hop routes is more quickly affected by moving nodes. For example, if a node at distance 1 moves away by 1 unit, the SNR change is 2^α, which causes the link to break (unless an unreasonably high SNR margin is applied). On the other hand, if a relay node is 3 units away moves by the same distance, the SNR change is only $(4/3)^\alpha$, which can probably be tolerated.

In general, the lifetime of a link is proportional to its (original) length. To show this, assume that routes are established with a certain SNR margin M_Θ, such that for a link of initial distance d, we have $\frac{P_0}{d^\alpha N} = \Theta\, M_\Theta$. At the critical distance $d + \Delta d$ when the link breaks, we have $\frac{P_0}{(d+\Delta d)^\alpha N} = \Theta$. Solving for Δd, we find

$$\Delta d = d(M_\Theta^{1/\alpha} - 1)\,. \tag{16}$$

For a given mobility pattern, Δd determines the lifetime of a link, so indeed longer links live proportionally longer.

I. Traffic Accumulation and Energy Balancing

For certain multihop networks such as sensor networks or multihop cellular networks, traffic accumulation around a base station (BS) or access point is a big problem. With strict short-hop routing, the relaying burden cannot be distributed among a high enough number of nodes, leading to a *critical area* around the BS whose nodes suffer from a short lifetime (see Fig. 2) The more nodes can reach the BS directly, the better distributed the load can be [15].

J. Variance in Hop Length

In random networks, due to the variance in hop length, the variance in energy consumption is large when nearest-neighbor schemes are used, causing substantial imbalance in the energy consumption. Assuming that a power control scheme is used that adapts the transmit power to maintain a certain reception probability p_r^N (see (2)). For an n-hop route, we are interested in the power consumption at the node

[4] Hence, after every convolution, the support needs to be scaled, which results in a reduction of the variance.

[5] The Gaussian approximation is very accurate even for small n. For $\phi = \pi/2$ and $n = 1$, *e.g.*, the precise value is $2\sqrt{2}/\pi \approx 0.9003$, whereas this approximation yields $e^{-\pi^2/96} \approx 0.9023$, so the error is only 0.2%. The second order Taylor expansions are identical, even for $n = 1$.

that has to transmit over the largest distance, since this node determines the lifetime of the route. From (1) we know that the distance to the nearest neighbor is Rayleigh distributed. So, for a path loss exponent $\alpha = 2$, the power level is given by an exponential distribution, and we need to determine the expected maximum of n exponentially distributed random variables. The same problem appears in selection combining of multiple transmission in fading channels, so we can apply the result from [16, p.316] that this maximum is given by the harmonic sum, *i.e.*,

$$\mathbb{E}[\max\{R_1,\ldots,R_M\}^2] = \sum_{k=1}^{M} \frac{1}{k} \gtrapprox \ln(M) + \gamma_{\mathrm{em}}\,, \qquad (17)$$

where $\gamma_{\mathrm{em}} \approx 0.577$ is the Euler-Mascheroni constant. The bound $\ln(M) + \gamma_{\mathrm{em}}$ is tight already for moderate M, so the expected maximum $\mathbb{E}[\max\{R_1,\ldots,R_M\}^2]$ is sharply lowerbounded by $\mathbb{E}[R^2](\ln M + \gamma_{\mathrm{em}})$. For $\alpha \geqslant 2$, with Jensen's inequality, we have $\mathbb{E}[(X^2)^{\alpha/2}] \geqslant \mathbb{E}[X^2]^{\alpha/2}$ and thus

$$\begin{aligned}\mathbb{E}[\max\{R_1,\ldots,R_M\}^\alpha] &\geqslant \left(\mathbb{E}[\max\{R_1,\ldots,R_M\}^2]\right)^{\alpha/2} \qquad (18)\\ &\gtrapprox \left(\mathbb{E}[R^2](\ln M + \gamma_{\mathrm{em}})\right)^{\alpha/2},\end{aligned}$$

or, since $\mathbb{E}[R^2] = (4/\pi)\mathbb{E}[R]^2$,

$$\mathbb{E}[\max\{R_1,\ldots,R_M\}^\alpha] \gtrapprox \left(\frac{2}{\sqrt{\pi}}\mathbb{E}[R]\right)^\alpha (\ln M + \gamma_{\mathrm{em}})^{\alpha/2}\,. \qquad (19)$$

For $\alpha = 1$ (the distance itself), we have the *upper* bound

$$\mathbb{E}[\max\{R_1,\ldots,R_M\}] \lessapprox \sqrt{\mathbb{E}[R^2](\ln M + \gamma_{\mathrm{em}})}\,. \qquad (20)$$

This shows that the maximum energy consumption over M nearest-neighbor hops grows with at least $\ln M$ in networks with random node distribtution. If longer hops are permitted, the distances and thus the power consumption can be better equalized among the nodes in a route.

K. Bounded Attenuation

As pointed out in the introduction, a path loss model with a singularity at distance $d = 0$ is not realistic for networks with high density. Clearly, there is a bound on the received power. If we assume that this bound is achieved for distances $0 < d < R$, then there is no benefit of using shorter hops than R, since the interference remains constant and the received power does not increase if the distance is decreased further. This problem is particularly relevant in dense networks [17].

L. Multicast Advantage

So far, we have only addressed unicast routing. For multicast, other tradeoffs between short-hop and long-hop routing exist. In particular, as discussed in [9], it is often advantageous for a source to transmit at high power levels to reach a maximum number of nodes in the multicast group.

IV. Conclusion

We have listed twelve compelling reasons why the tradeoff between routing over many short hops and routing over fewer longer hops is not as clear as it is often assumed. Not all reasons apply to all types of networks, of course, but several of them will be relevant for most networks. The conclusion is that *routing as far as possible is a very competitive strategy* in many cases. Conversely, from a design perspective, the peak transmit power should be chosen such that a node can reach well beyond nearest neighbors.

References

[1] J. Jun and M. L. Sichitiu, "The Nominal Capacity of Mesh Networks," *IEEE Wireless Communications*, vol. 5, pp. 8–14, Oct. 2003.

[2] Y.-D. Lin and Y.-C. Hsu, "Multihop Cellular: A New Architecture for Wireless Communications," in *IEEE INFOCOM'00*, vol. 3, pp. 1273–1282, 2000.

[3] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, Mar. 2000.

[4] E. S. Sousa and J. A. Silvester, "Optimum Transmission Ranges in a Direct-Sequence Spread-Spectrum Multihop Packet Radio Network," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 762–771, June 1990.

[5] R. Meester and R. Roy, *Continuum Percolation*. Cambridge University Press, 1996.

[6] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," *IEEE Personal Communications*, vol. 6, pp. 46–55, Apr. 1999.

[7] C. E. Perkins, ed., *Ad Hoc Networking*. Addison Wesley, 2000. ISBN 0-201-30976-9.

[8] M. Haenggi, "On Routing in Random Rayleigh Fading Networks," *IEEE Transactions on Wireless Communications*, 2004. Accepted for publication. Available at `http://www.nd.edu/~mhaenggi/routing.pdf`.

[9] A. Ephremides, "Energy Concerns in Wireless Networks," *IEEE Wireless Communications*, vol. 9, pp. 48–59, Aug. 2002.

[10] M. Sikora, J. N. Laneman, M. Haenggi, D. J. Costello, and T. Fuja, "On the Optimum Number of Hops in Linear Ad Hoc Networks," in *IEEE Information Theory Workshop (ITW'04)*, (San Antonio, TX), Oct. 2004. Submitted for publication.

[11] M. Haenggi, "The Impact of Power Amplifier Characteristics on Routing in Random Wireless Networks," in *IEEE Global Communications Conference (GLOBECOM'03)*, (San Francisco, CA), Dec. 2003. Available at `http://www.nd.edu/~mhaenggi/globecom03.pdf`.

[12] F. Xue and P. R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks," *Wireless Networks*, 2004. To appear. Available at `http://black.csl.uiuc.edu/~prkumar/ps_files/connect.pdf`.

[13] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Transactions on Information Theory*. Accepted for publication. Available at: `http://www.nd.edu/~jnl/pubs/it2002.pdf`.

[14] M. Haenggi, "Analysis and Design of Diversity Schemes for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*. Accepted for publication. Available at `http://www.nd.edu/~mhaenggi/jsac_adhoc.pdf`.

[15] M. Haenggi, "Energy-Balancing Strategies for Wireless Sensor Networks," in *IEEE International Symposium on Circuits and Systems (ISCAS'03)*, (Bangkok, Thailand), May 2003. Available at `http://www.nd.edu/~mhaenggi/iscas03.pdf`.

[16] W. C. Jakes (Ed.), *Microwave Mobile Communications*. IEEE Press, 1993. ISBN 0-7803-1069-1.

[17] O. Dousse and P. Thiran, "Connectivity vs Capacity in Dense Ad Hoc Networks," in *IEEE INFOCOM*, (Hongkong), Mar. 2004.

Analysis and Design of Diversity Schemes for Ad Hoc Wireless Networks

Martin Haenggi, *Senior Member, IEEE*

***Abstract*—Diversity schemes permit efficient communication over fading channels but are often hard to analyze and design in networks with many nodes. For Rayleigh-fading channels, there exists an interesting relationship between resistive circuits and time and path diversity mechanisms in wireless ad hoc networks. A resistor-like network element, the *erristor*, representing the normalized noise-to-signal ratio, is introduced. Given an end-to-end packet delivery probability, the logarithmic mapping from link reception probabilities to erristor values greatly simplifies the problems of power allocation and the selection of time and path diversity schemes, which is illustrated in a number of examples. We focus on transmission strategies with selection combining and simple noncoherent "decode-and-forward" strategies, which is motivated by their practicality. Thanks to its conceptual simplicity, the formalism that is developed provides valuable insight into the benefits of diversity mechanisms.**

***Index Terms*—Ad hoc networks, diversity methods, Rayleigh channels.**

I. Introduction

IN AD HOC wireless networks, energy and interference considerations often necessitate *multihop routing*, where nodes also act as routers, forwarding other nodes' packets [1], [2]. Routing schemes that were developed for wired networks will perform suboptimally since they are based on virtually error-free point-to-point links, thereby ignoring two fundamental properties of the wireless link: 1) the fragility of the channel due to fading and interference [2], [3] and 2) the inherent broadcast property of wireless transmissions.[1] Fading can be an advantage if transmissions can be scheduled opportunistically [4], [5], and the broadcast property can be exploited by transmission schemes that are based on the principle of *cooperative diversity* [6], where nodes coordinate both direct and relayed transmissions. Cooperative diversity is a form of spatial diversity, which, in the case of static single-antenna nodes, reduces to path diversity. The other promising strategy (in the case of narrowband channels) against fading is time diversity, which, for slow (nonergodic) fading channels and relatively short packets, is mainly exploited in the form of retransmissions.

For the analysis of multihop packets networks, a deterministic "disk model" is often used [7]–[15], where the radius for a successful transmission has a power-dependent deterministic value, irrespective of the condition of the wireless channel. Interference is commonly taken into account using the same geometric disk abstraction. The stochastic nature of the fading channel and, thus, the fact that the signal-to-noise-and-interference ratio (SINR) is a random variable are neglected. In general, the volatility of the channel cannot be ignored in wireless networks [2], [3], and a rigorous assessment of diversity benefits is not possible with a deterministic model. The inaccuracy of the "prevalent all-or-nothing model," as the disk model is called in [16], has also been pointed out in [17] and is easily demonstrated experimentally [18], [19]. To overcome some of these limitations of the disk model, we employ a Rayleigh-fading link model that relates transmit power, large-scale path loss, and the success of a transmission.

Based on this model, we present a simple yet powerful formalism that allows an efficient analysis and design of time and path diversity strategies with selection combining for Rayleigh-fading channels. In the analysis problem, the transmit power levels are given and the end-to-end reliability p_{EE} is to be determined, whereas in the (more interesting) design problem, we assume that the application dictates a certain end-to-end reliability p_{D}, and the question is how to choose the transmit powers, the relays (paths), and the number of transmissions over each link in order to minimize energy consumption and/or maximize network lifetime under the constraint $p_{\mathrm{EE}} \geq P_{\mathrm{D}}$. We focus on noncoherent "decode-and-forward" strategies, which are, in contrast to coherent combining, straightforward to implement. Nonetheless, they exhibit greatly superior performance to schemes without diversity, in particular at high SNR.

The rest of this paper is organized as follows. In Section II, the link model is presented. Section III introduces the "erristor" framework, where transmissions are characterized by a resistor-like circuit element whose value equals the normalized noise-to-signal ratio at the receiver, and discusses its use for multihop routes and time diversity schemes. Section IV focuses on path diversity schemes and presents several examples, and Section V concludes the paper.

II. Link Model

We assume a narrowband multipath channel, modeled as a Rayleigh block fading channel [20] with an additive white Gaussian noise (AWGN) process z. The received signal at time k is $y_k = a_k x_k + z_k$, where a_k is the large-scale path loss multiplied by the fading coefficient. The variance of the noise process is denoted by N_0.

The transmission from node i to node j is successful if the SINR γ is above a certain threshold Θ that is determined by

Manuscript received October 15, 2003; revised July 16, 2004. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA)/IXO under Grant AF-F30602-01-2-0526 and in part by the National Science Foundation (NSF) under Grant ECS02-25265 and Grant ECS03-29766.

The author is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: mhaenggi@nd.edu).

Digital Object Identifier 10.1109/JSAC.2004.837357

[1] We assume that omnidirectional antennas are employed.

the communication hardware and the modulation and coding scheme [2].

With the assumptions above, the received power Q is exponentially distributed with mean $\bar{Q} = P_0 d^{-\alpha}$, where P_0 is proportional to the transmit power,[2] $d = \|x_i - x_j\|_2$ the distance between transmitter and receiver, and α the large-scale path loss exponent. I is the interference power affecting the transmission, i.e., the sum of the received powers of all the undesired transmitters.

The following theorem shows that for Rayleigh-fading networks, it is possible to analyze noise and interference separately.

Theorem 1: In a Rayleigh-fading network, the reception probability $p_r = \mathbb{P}[\gamma \geq \Theta]$ can be factorized into the reception probability of a zero-interference network p_r^N and the reception probability of a zero-noise network p_r^I as follows:

$$p_r = \underbrace{\exp\left(-\frac{\Theta N_0}{P_0 d_0^{-\alpha}}\right)}_{p_r^N} \cdot \underbrace{\prod_{i=1}^{k} \frac{1}{1 + \Theta \frac{P_i}{P_0}\left(\frac{d_0}{d_i}\right)^{\alpha}}}_{p_r^I} \tag{1}$$

where d_0 is the distance of the desired transmitter, $d_i, 1 \leq i \leq k$ are the distances of the k interferers, and P_i are their transmit power levels.

Proof: Let Q_0 denote the received power from the desired source and $Q_i, i = 1, \ldots, k$, the received power from k interferers. All the received powers are exponentially distributed, i.e., $p_{Q_i}(q_i) = 1/\bar{Q}_i e^{-r_i/\bar{Q}_i}$, where $\bar{Q}_i$ denotes the average received power $\bar{Q}_i = P_i d_i^{-\alpha}$. Conditioned on I, the probability of correct reception is simply an exponential, so we have

$$\begin{aligned} p_r &= \mathbb{E}_I\left[\mathbb{P}[Q_0 \geq \Theta(I + N_0) | I]\right] \\ &= \mathbb{E}_I\left[\exp\left(-\frac{\Theta(I + N_0)}{\bar{Q}_0}\right)\right] \\ &= \int_0^\infty \cdots \int_0^\infty \exp\left(-\frac{\Theta\left(\sum_{i=1}^k q_i + N_0\right)}{\bar{Q}_0}\right) \\ &\quad \times \prod_{i=1}^{k} p_{Q_i}(q_i)\, \mathrm{d}q_1 \cdots \mathrm{d}q_k \end{aligned}$$

from which (1) follows.[3] ■

This allows an independent analysis of the effect caused by noise and the effect caused by interference. Since we are concerned with the fundamental energy benefits of diversity, the focus of this paper is the noise, i.e., on the first factor in (1). Clearly, p_r^I is invariant under (global) power scaling, so it does not give an indication at which power levels to transmit. The results that are derived are directly applicable to the case of light traffic ($p_r^I \to 1$). For the heavy traffic case, a separate interference analysis would have to be carried out, as in [22] and [23], and the resultant p_r^I would have to be multiplied by the p_r^N derived here.

In a zero-interference network, the reception probability over a link of distance d at a transmit power P_0, is given by

$$p_r := \mathbb{P}[\gamma_N \geq \Theta] = e^{-\frac{\Theta N_0}{P_0 d^{-\alpha}}}. \tag{2}$$

Solving for P_0, we get for the necessary transmit power to achieve p_r

$$P_0 = \frac{d^\alpha \Theta N_0}{-\ln p_r}. \tag{3}$$

Note that for $\alpha \to \infty$, the fading model turns into a disk model. For $d < 1$, the reception probability is one, for $d > 1$, it is zero. Similarly, for the interference term in (1), if all the interferers are further away than the desired transmitter, i.e., $d_0 < \min\{d_1, \ldots, d_k\}$, reception is guaranteed. Otherwise, reception is impossible.

III. Erristor Representation

A. Connections Without Retransmission

Over an n-hop connection from node 0 to node n in an ad hoc network, the reception probability (end-to-end reliability) is

$$p_{\mathrm{EE}} = e^{-\Theta \sum_{i=1}^{n} \frac{1}{\bar{\gamma}_i}} \tag{4}$$

where $\bar{\gamma}_i$ denotes the mean SNR at receiver i. Let R denote the normalized average noise-to-signal ratio (NSR) at the receiver, i.e., $R := \Theta/\bar{\gamma}$. We get

$$-\ln p_{\mathrm{EE}} = \sum_{i=1}^{n} R_i = R_{\mathrm{tot}}. \tag{5}$$

With p_{D} the desired end-to-end reliability, the condition $p_{\mathrm{EE}} \geq p_{\mathrm{D}}$ translates into the condition that the sum or the *series connection*[4] of the NSR values R_i is at most $R_{\mathrm{D}} := -\ln p_{\mathrm{D}}$. So, the individual R_i's can be replaced by an equivalent R_{tot}. For a single link, we have

$$R = -\ln p_r \iff p_r = e^{-R}. \tag{6}$$

For probabilities close to 1 (or $R \ll 1$), the following first-order approximations are accurate

$$\hat{R} := 1 - p_r \gtrsim R \iff \hat{p}_r := 1 - R \lesssim p_r. \tag{7}$$

This approximation shows that for small values, the NSR can be considered equivalent to the packet loss probability. To emphasize this fact and the resistor-like series connection property of the NSR, we denote R as an "erristor" and its value as its "erristance." It follows from (5) that over a multihop connection, the noise accumulates and the packet loss probabilities simply add up.

[2] This equation does not hold for very small distances. So, a more accurate model would be $\bar{Q} = P_0' \cdot (d/d_0)^{-\alpha}$, valid for $d \geq d_0$, with P_0' as the average value at the reference point d_0, which should be in the far field of the transmit antenna. At 916 MHz, for example, the near field may extend up to 3–4 ft (several wavelengths).

[3] A similar calculation has been carried out in [21, Appendix] for a network with spreading gain and equal transmit powers for all nodes.

[4] In terms of the SNR values, this corresponds to a parallel connection, which was pointed out in [24].

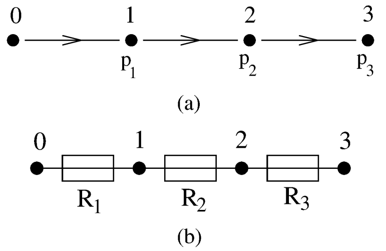


Fig. 1. (a) Three-hop connection with link reception probabilities p_1, p_2, and p_3 and the erristor circuit. (b) The erristor values are the normalized noise-to-signal ratios R_1, R_2, and R_3. $R_i = -\ln p_i$.

The relationship between the transmit power and R is

$$P_0 = d^\alpha \bar{\gamma} N_0 = \frac{d^\alpha \Theta N_0}{R}. \tag{8}$$

Henceforth, $P := d^\alpha / R$ denotes the normalized (by ΘN_0) transmit power.

Example 1: Fig. 1(a) shows an example with three links and their reception probabilities. From (5), we know that a series of hops translates into a series connection of erristors, hence, we find the corresponding erristor network in Fig. 1(b). For $p_{\mathrm{D}} = 90\%$, for example, the value of $R_1 + R_2 + R_3$ must be at most $R_D = -\ln p_{\mathrm{D}} \approx 0.105$. If all the power levels are equal, this can be achieved by setting $R_1 = R_2 = R_3 = 0.105/3 = 0.035$. A possible solution with unequal power is $R_1 = R_2 = 0.05$ and $R_3 = 0.005$. In this case, the probability after two links is $e^{-0.05}e^{-0.05} \approx 90.5\%$, which is already close to 90%. Consequently, a high amount of energy is consumed at the third link to ensure packet reception with the required probability $e^{-0.005} = 99.5\%$.

If the internode distances d_i (between node $i-1$ and node i) are given, a solution can be determined that ensures that all the *transmit* power levels have the same value P. From $(d_1^\alpha + d_2^\alpha + d_3^\alpha)/P = -\ln p_{\mathrm{D}}$, we get

$$P = \frac{d_1^\alpha + d_2^\alpha + d_3^\alpha}{-\ln p_{\mathrm{D}}}. \tag{9}$$

For $d_i = i, \alpha = 2$, and $p_{\mathrm{D}} = 90\%$, for example, we get $P \approx 14 \cdot 9.5 = 133$ and $R_1 \approx 0.0075, R_2 \approx 0.03$, and $R_3 \approx 0.0677$. □

B. Connections With Time Diversity (Retransmissions)

In general, with n transmissions over the same link at NSR levels R_i and selection combining, the total reception probability is

$$p_n = 1 - \prod_{i=1}^{n} \left(1 - e^{-R_i}\right). \tag{10}$$

To derive a general rule for the simplification of these expressions, we apply the following theorem.

Theorem 2: For $(x_1, x_2, \ldots, x_n) \in (\mathbb{R}_0^+)^n$

$$1 - \prod_{i=1}^{n} \left(1 - e^{-x_i}\right) \geq e^{-\prod_{i=1}^{n} x_i} \tag{11}$$

with equality if and only if $\prod_{i=1}^{n} x_i = 0$.

The proof is presented in the Appendix.

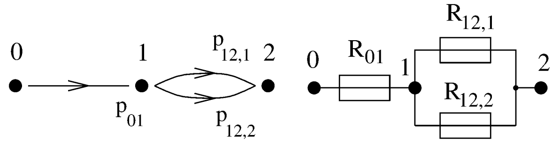


Fig. 2. (Left) A two-hop connection with two transmissions over the second hop. (Right) Corresponding erristor circuit.

So, erristors connected in series behave like regular resistors, whereas the values of erristors connected in parallel (retransmissions) have to be multiplied. Due to the bound (11), the resultant end-to-end reliability will be slightly higher than the one specified.

Example 2: Consider the simple example in Fig. 2. A packet is transmitted with reception probability p_{01} over the first hop and transmitted twice over the second hop, with probabilities $p_{12,1}$ and $p_{12,2}$, respectively. The end-to-end reliability of the connection is $p_{\mathrm{EE}} = p_{01} \cdot (1 - (1 - p_{12,1})(1 - p_{12,2}))$. Let $p_{\mathrm{D}} = 90\%$. How to allocate the transmit powers?

For the second link, we have $p_{12} = 1 - (1 - p_{12,1})(1 - p_{12,2}) = 1 - (1 - e^{-R_{12,1}})(1 - e^{-R_{12,2}})$. From Theorem 2, $e^{-R_{12,1}R_{12,2}}$ is a lower bound for p_{12}, and for $R_1 \ll 1$ and $R_2 \ll 1$, the bound is tight. Thus, we may replace $R_{12,1}$ and $R_{12,2}$ by $R_2 = R_{12,1}R_{12,2}$. In the erristor diagram, the two transmissions are illustrated by a parallel connection (see Fig. 2). From $R_1 + R_2 < R_D = 0.1$, a possible choice is $R_1 = R_2 = 0.05$ and $R_{12,1} = R_{12,2} = \sqrt{5}/10$. Thanks to the retransmission, node 2 consumes less than half the energy of node 1, assuming the distances are the same. An allocation where both nodes use the same power is $R_1 = \sqrt{110}/5 - 2 \approx 0.098$, and $R_{12,1} = R_{12,2} = 2R_1$. This way, node 1 benefits from node 2's retransmission and uses almost only half the power. □

For n transmissions with the same power level R, the difference between the precise probability value $1 - (1 - e^{-R})^n$ and the lower bound e^{-R^n} is plotted in Fig. 3. The erristance threshold where the bound is within 1% is $R = 0.236$ for $n = 2$ and $R = 0.375$ for $n = 4$. Thus, for $R < 1/4 (p > 78\%)$, the bound is sufficiently tight for all practical purposes. For values $R > 1$, the bound is loose, and the multiplication would not make sense, since the overall erristance increases, although, of course, even a retransmission with low power still leads to an improvement in the total link reception probability. However, for $R \gtrapprox 1/2$, a single transmission outperforms splitting the power into two transmissions. For two transmissions at NSR $2R$, the reception probability is $p_r = 1 - (1 - e^{-2R})^2$, whereas for a single transmission at NSR R, we get $p_r' = e^{-R}$. The two probabilities are equal for

$$R = \ln 2 - \ln\left(\sqrt{5} - 1\right) \approx 0.48. \tag{12}$$

So, for $R \gtrapprox 1/2$, the reception probability is higher for a single transmission at NSR R. From a practical point of view, reception probabilities lower than 37% may not be worth spending energy at the receiver. By excluding erristances greater than one, this is taken into account, and, at the same time, the validity of the multiplication property is preserved.

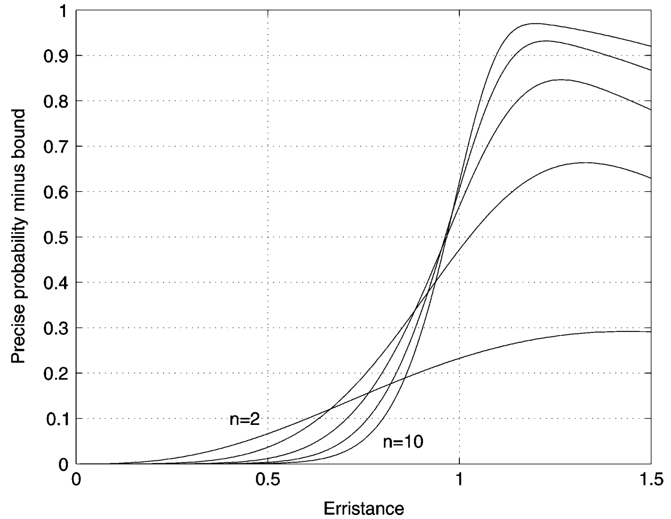


Fig. 3. Difference between the exact probability and the lower bound for $n = 2, 4, 6, 8, 10$ transmissions with equal erristance R.

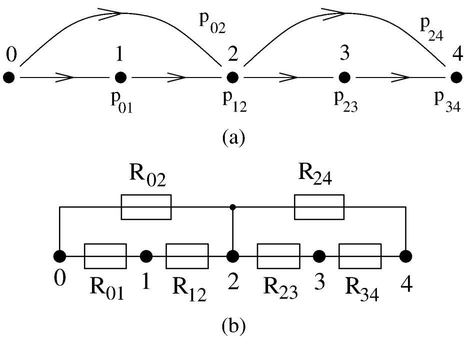


Fig. 4. (a) Network that exploits path diversity and (b) its erristor circuit.

IV. Path Diversity

A. Transmissions Over Independent Paths

With the addition and multiplication property of the erristors, we have all the ingredients to analyze and design path diversity schemes.

Example 3: Fig. 4 displays an example of a network, where path independence is guaranteed even when the channels have a long coherence time. By conventional analysis, the end-to-end reception probability is

$$p_{\rm EE} = (1-(1-e^{-R_{01}}e^{-R_{12}})(1-e^{-R_{02}})) \cdot (1-(1-e^{-R_{23}}e^{-R_{34}})(1-e^{-R_{24}})). \qquad (13)$$

How to choose the R_{ij}'s to guarantee $p_{\rm EE} \geq p_{\rm D} = 95\%$? The equivalent erristance [Fig. 4(b)] is $R_{\rm tot} = (R_{01}+R_{12})R_{02} + (R_{23}+R_{34})R_{24}$, and $p_{\rm EE} = e^{-R_{\rm tot}}$. For $p_{\rm D} = 95\%$, we have $R_{\rm tot} \approx 0.05$. Thanks to the symmetry, $(R_{01}+R_{12})R_{02} = (R_{23}+R_{34})R_{24} = 0.025$ is a solution; hence, we may set nearest-neighbor hops to $R_{i-1,i} = 0.05$ and $R_{02} = R_{24} = 0.25$. Note that the value for the two longer hops is five times bigger, which means that the necessary *transmit* powers are comparable if the nodes have equal distances and the path loss exponent is between 2 and 3. So, the diversity scheme and the power allocation guarantee $p_{\rm EE} = e^{-0.05} \approx 95.1\% > p_{\rm D}$. The conventional analysis (13) yields $p_{\rm EE} \approx 95.8\%$ which is, as expected, slightly larger. The formalism also permits a rapid reallocation of resources, if necessary. Assume node 3 runs out of energy. With $R_{34} \rightarrow \infty$, we see immediately that R_{23} becomes useless, and the only path in the lower half of the diagram will be the one with R_{24}. What value does R_{24} need to have to ensure $p_{\rm D}$? Without changing the other erristances, we find $R_{24} = 0.025$. □

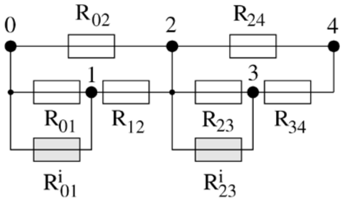


Fig. 5. Erristor circuit for example 3 (see Fig. 4) including implicit transmissions. Implicit erristors are shaded gray.

The total normalized energy consumption (per packet) at each node can easily be determined

$$E_i = \sum_{j=1}^{m} \frac{d_{ij}^{\alpha}}{R_{ij}} \qquad (14)$$

where m is the number of outgoing paths from node i.

B. Implicit Transmissions

In example 2 (Fig. 2), if node 2 listens to the transmission from node 0 to node 1, then this *implicit transmission* has to be modeled by an additional erristor for an accurate analysis. This implicit erristor is free in terms of transmit power (but still requires power to receive the packet); it represents the benefit from the wireless medium, often denoted as the *wireless multicast advantage* [2], [25].

Example 2 (Continued): Assume $p_{\rm D} = 99\%$, so $R_{\rm tot} \approx 0.01$. This is achieved by setting $R_{01} = 0.005$ and $R_{12,1} = R_{12,2} = 0.07$. However, since there is an implicit transmission from 0 to 2, there is an erristor in parallel with a value of $R_{02}^i = R_{01}(d_{02}/d_{01})^\alpha$ (the superscript i indicates an implicit transmission). Assuming $d_{02} = 2d_{01}$ and $\alpha = 3$, we get $R_{02}^i = 0.04, R_{\rm tot} = 0.01 \cdot 0.04$, and $p_{\rm EE} \approx 99.96\%$, which is much better than the target of 99%. So we can reduce R_{01} to a value that guarantees $(R_{01}+0.005)R_{01} \cdot 8 = 0.01$. Solving the resulting quadratic equation yields $R_{01} \approx 1/30$, which corresponds to less than 1/6 of the original power. □

For large path loss exponents and/or small transmit power levels, the benefit to listeners that are farther away than the intended receiver becomes small, since the implicit erristances will be close to one or even above. However, if the implicit receiver is closer than the intended one or if the transmit power is relatively high, it is worthwhile having the nodes awake and listening.

Example 3 (Continued): In example 3 (Fig. 4), there is an implicit transmission from node 2 to node 3 when node 2 is transmitting to node 4. If node 3 ignores this transmission, then the previous analysis was correct. If it takes advantage of that information, we have to add another erristor R_{23}^i, as shown in Fig. 5. Analogously, R_{01}^i models the implicit transmission $\overrightarrow{01}$ that occurs due to the explicit transmission $\overrightarrow{02}$.

Assuming equal distances between neighboring nodes, $R_{01}^i = 2^{-\alpha}R_{02}$ and $R_{23}^i = 2^{-\alpha}R_{24}$. For $\alpha = 2$ and using the

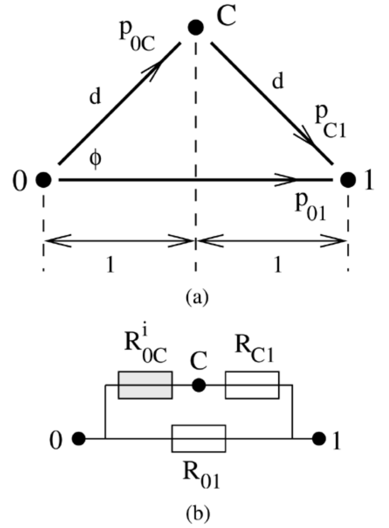


Fig. 6. (a) Simple cooperative scheme and (b) the corresponding erristor circuit. Node C is relaying a packet from node 0 to assist node 0. The distances $\overline{0C}$ and $\overline{C1}$ are $d = 1/\cos\phi$.

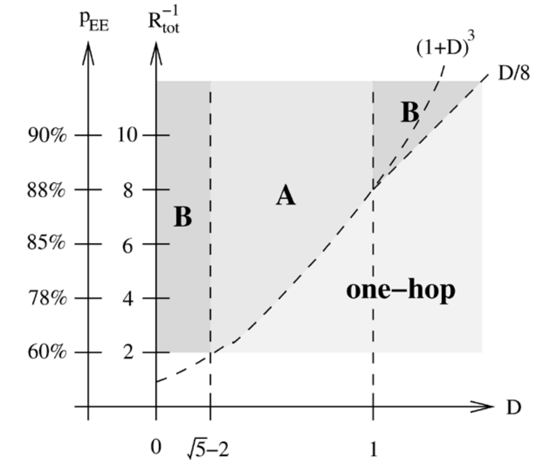


Fig. 7. Visualization of the regions in the $(D, R_{\mathrm{tot}}^{-1})$ plane where the different strategies are optimum. Note that the curves $D/8$ and $(1+D)^3$ intersect at the points $(\sqrt{5}-2, (\sqrt{5}-1)^3)$ and $(1, 8)$.

same values as before, $R_{i,i-1} = 0.05$, and $R_{02} = R_{24} = 0.25$, we find $R^i_{01} = R^i_{23} = 0.0625$ and

$$R_{\mathrm{tot}} = 2\left(\frac{1}{20}\cdot\frac{1}{16}+\frac{1}{20}\right)\cdot\frac{1}{4} \approx 0.027 \tag{15}$$

resulting in $p_{\mathrm{EE}} \approx 97.3\%$, which is larger than the target of 95%. Considering that $1/16 \ll 1$, we may try to omit the explicit transmission $\overrightarrow{01}$ completely, which results in $R_{\mathrm{tot}} = (1/16+1/20)/2 \approx 0.056$ and $p_{\mathrm{EE}} = 94.4\%$. A slight decrease of R_{02} and R_{24} by 10% each brings p_{EE} to 95.2%. □

The next example demonstrates how the erristor formalism transforms complex logarithmic relationships into simple polynomial ones, which are analytically tractable.

Example 4. A Simple Cooperative Scheme: In Fig. 6, a situation is shown where node 0 wants to transmit to node 1, and the cooperative node C may help as a relay. From the erristor circuit, it can be seen that there is no explicit transmission from 0 to C. The goal is to determine which transmission strategy minimizes the total transmit energy E_{tot} given a certain total erristance R_{tot}.

Let $D := (d/2)^\alpha$. With $R_{\mathrm{tot}} = R_{01}(R^i_{0C}+R_{C1})$ and $R^i_{0C} = DR_{01}$, we get $R_{\mathrm{tot}} = R_{01}(R_{01}D+R_{C1})$ and $E_{\mathrm{tot}} = 2^\alpha/R_{01}+d^\alpha/R_{C1} = 2^\alpha(R_{01}^{-1}+DR_{C1}^{-1})$.

Strategy A—Equal Received Power: A possible strategy is to have C transmit at a power level that makes the received power at node 1 equal to that from the direct transmission $\overrightarrow{01}$, i.e., $R_{C1} = R_{01} =: R$. R_{tot} simplifies to $R_{\mathrm{tot}} = R^2(1+D)$ and, thus, $R = \sqrt{(R_{\mathrm{tot}})/(1+D)}$. Inserting this expression into the transmit energy $E_{\mathrm{tot}} = (2^\alpha)/(R)(1+D)$ yields

$$E^A_{\mathrm{tot}} = \frac{2^\alpha(1+D)^{3/2}}{\sqrt{R_{\mathrm{tot}}}}. \tag{16}$$

Strategy B—Equal Transmit Power: Here, we assume that both node 0 and C use the same transmit power. With $R := R_{01}$ and $R_{C1} = R^i_{0C} = RD$, we have $R_{\mathrm{tot}} = 2DR^2$ and $R = \sqrt{(R_{\mathrm{tot}})/(2D)}$. The total energy consumption is simply $E_{\mathrm{tot}} = 2\cdot 2^\alpha/R$, or, as a function of R_{tot}

$$E^B_{\mathrm{tot}} = 2^{\alpha+1}\sqrt{\frac{2D}{R_{\mathrm{tot}}}}. \tag{17}$$

The energy consumption ratio of strategies A and B is

$$\rho := \frac{E^B_{\mathrm{tot}}}{E^A_{\mathrm{tot}}} = \frac{2\sqrt{2D}}{(1+D)^{3/2}}. \tag{18}$$

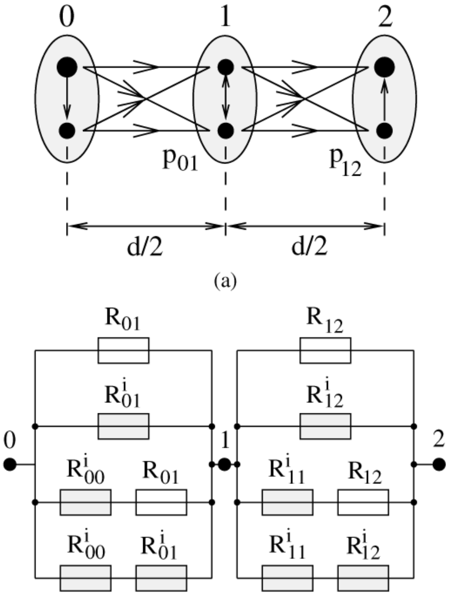
$\rho = 1$ for $D = 1$ and $D = \sqrt{5}-2 \approx 0.236$. For $\sqrt{5}-2 < D < 1$, strategy A is preferable ($\rho > 1$). The maximum ρ, however, is only $(4/9)\sqrt{6} \approx 1.089$, occurring at $D = 1/2$. So, strategy A is at most 8.9% better.

To get a complete view, we also discuss the case of a direct one-hop transmission and an explicit two-hop scheme without a direct path from 0 to 1. For the one-hop case, we have $E^{\mathrm{one}}_{\mathrm{tot}} = 2^\alpha/R_{\mathrm{tot}}$, and for the two-hop case (assuming equal transmit powers), there are two erristors in series with value $R_{\mathrm{tot}}/2$ and, thus, $E^{\mathrm{two}}_{\mathrm{tot}} = 4d^\alpha/R_{\mathrm{tot}} = 4DE^{\mathrm{one}}_{\mathrm{tot}}$.

The one-hop strategy is better if $4D \geq 1$, or, in terms of the actual distance $d, d > 2^{1-2/\alpha}$. So, for $\alpha = 2$ and for $d \geq 2$, one-hop is always better, even for $\alpha \to \infty$, which is easily explained, since node C is then not closer than node 1. As a function of the angle $\phi = \arccos(1/d)$, the condition for one-hop to be better is expressed as $\phi \geq \arccos(2^{2/\alpha-1})$. For $\alpha \to \infty$, the critical angle is $\phi = \pi/3$ (corresponding to an equilateral triangle 0C1), as expected.

The last step is the comparison of these simple schemes with the cooperative strategies A and B. First, we note that B always outperforms the two-hop scheme, since it exploits "free" information that is transmitted over the direct path. The tournament between A and one-hop is won by A if $R_{\mathrm{tot}} < (1+D)^{-3}$, and B wins against one-hop if $R_{\mathrm{tot}} < (8/D)$. The resulting division of the $(D, R_{\mathrm{tot}}^{-1})$-plane in the different strategies is shown in Fig. 7. Since R_{tot}^{-1} is related to the SNR, Fig. 7 suggests that cooperation is beneficial at high SNR and relatively small distances D, in agreement with intuition. □

Example 5. Virtual Antenna Arrays: Several nodes that are close may cooperate and act as a virtual antenna array, exploiting spatial diversity. The performance of such arrays was analyzed in [26] from an information-theoretic perspective.



(a)

(b)

Fig. 8. (a) Virtual antenna scheme and (b) its erristor circuit.

Here, not assuming any form of lower-layer coordination within the cluster, we apply the erristor formalism to compare these schemes with conventional multihop routing. Fig. 8 shows the erristor diagram of a simple scenario with two nodes assisting each other at the source, in the middle, and at the destination. So, instead of individual nodes, we have clusters of two nodes at positions 0, 1, and 2. It is assumed that the intracluster distances are much smaller than the intercluster distances $d/2$. When the source node in cluster 0 is transmitting to cluster 1, his peer will receive that packet with probability (almost) one since $R_{00}^i \ll R_{01}$. In the next time slot, this peer node will transmit the same packet to cluster 1. Hence, the same packet is delivered over four different paths. Similarly, cluster 1 relays the packet to cluster 2 over four paths. In the case that the actual destination node itself in cluster 2 does not correctly receive the packet, an additional short intracluster transmission is required, whose energy is neglected in the following analysis.

We assume $R_{01} = R_{12} := R$. For the diversity scheme, with $R_{00}^i \ll R$, we get $R_{\text{tot}} = 2R^4$ and

$$E_{\text{tot}} = 4\frac{(d/2)^\alpha}{R} = 4\left(\frac{d}{2}\right)^\alpha \left(\frac{2}{R_{\text{tot}}}\right)^{\frac{1}{4}}. \tag{19}$$

For comparison, for a four-hop connection with hops of length $d/4$, we have $R_{\text{tot}} = 4R'$ and

$$E'_{\text{tot}} = 4\left(\frac{d}{4}\right)^\alpha \frac{4}{R_{\text{tot}}} = 2^{4-\alpha}\left(\frac{d}{2}\right)^\alpha R_{\text{tot}}^{-1}. \tag{20}$$

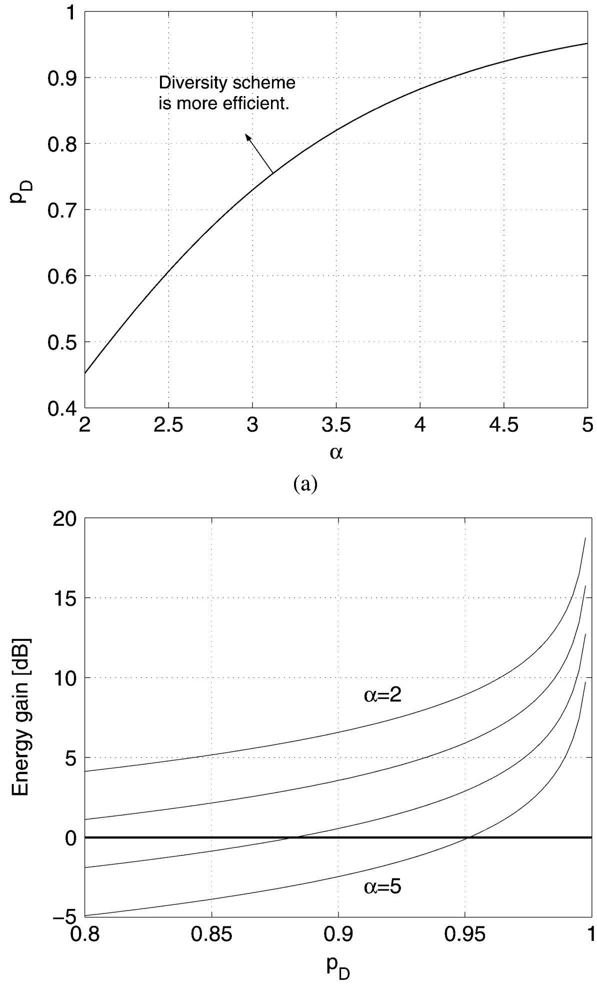
The ratio between the two energies is

$$\frac{E_{\text{tot}}}{E'_{\text{tot}}} = 2^{\alpha-\frac{7}{4}} R_{\text{tot}}^{\frac{3}{4}}. \tag{21}$$

Hence, the diversity scheme is more efficient for

$$R_{\text{tot}} < 2^{\frac{-4\alpha+7}{3}} \quad \text{or} \quad p_{\text{D}} > \exp\left(-2^{\frac{-4\alpha+7}{3}}\right). \tag{22}$$

This curve is plotted in Fig. 9(a). Substantial energy gains are possible for high p_{D} [see Fig. 9(b)]. When the path loss exponent increases by one, the energy gain decreases by 3 dB.



(a)

(b)

Fig. 9. (a) Region in the (α, p_{D}) plane where the diversity scheme outperforms conventional multihop routing. (b) Energy gain as a function of the end-to-end probability.

This diversity scheme can be generalized to clusters of size m that transmit over n hops. In this case, $R_{\text{tot}} = nR^{m^2}$ and

$$E_{\text{tot}} = mn\left(\frac{d}{n}\right)^\alpha \left(\frac{n}{R_{\text{tot}}}\right)^{\frac{1}{m^2}}. \tag{23}$$

For the multihop scheme with mn hops,[5] $R_{\text{tot}} = mnR'$ and

$$E'_{\text{tot}} = mn\left(\frac{d}{mn}\right)^\alpha \frac{mn}{R_{\text{tot}}}. \tag{24}$$

The ratio is

$$\frac{E_{\text{tot}}}{E'_{\text{tot}}} = R_{\text{tot}}^{1-\frac{1}{m^2}} n^{\frac{1}{m^2}-1} m^{\alpha-1} \tag{25}$$

from which we see that the energy gain is maximized for $m = 2$ (except for $\alpha = 2$, where $m = 3$ performs slightly better) and increases almost linearly in n. We conclude that for high p_{D} and α not too big, the diversity scheme clearly outperforms conventional multihop routing. □

[5] This comparison is fair both in terms of the number of nodes involved and in the delay, since the total number of transmissions is mn for both schemes.

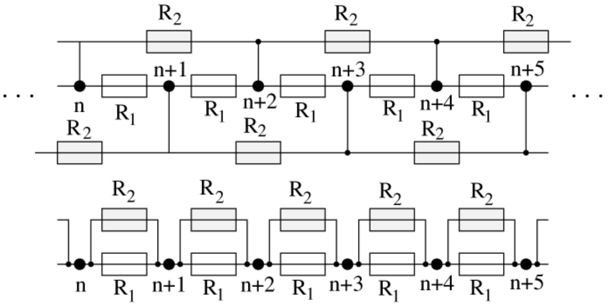


Fig. 10. (Top) Six nodes of a two-level leap-frog scheme. Every nodes listens to the transmissions of the two previous nodes. (Bottom) A conservative simplified erristor circuit.

Example 6. Long Line Networks with Equidistant Nodes: Consider a long line network with equidistant nodes of distance $d = 1$, with the source at the left end and the destination at the right end. Every node transmits once but listens to the transmissions of m of its left neighbors, resulting in $m - 1$ implicit transmissions. The corresponding erristor circuit for $m = 2$ is shown in Fig. 10. Due to the long chain of dependencies, such a leap-frog scheme[6] is difficult to analyze; we, therefore, simplify it by attaching the implicit transmission to node $n + 1$ rather than node n (Fig. 10). This yields an upper bound on the erristance, since the accumulated erristance at node n is certainly smaller than at node $n + 1$. The per-hop erristance R is then

$$R = R_1 \cdot 2^\alpha R_1 \cdot 3^\alpha R_1 \cdot \ldots \cdot m^\alpha R_1 = R_1^m (m!)^\alpha. \qquad (26)$$

The choice $m = \lfloor R_1^{-1/\alpha} \rfloor$ ensures that nodes do not listen to transmissions with low reliability. The benefit compared with conventional multihop routing is

$$\frac{R_1}{R} = R_1^{1-m} \cdot (m!)^{-\alpha} \qquad (27)$$

which is enormous, in particular at high SNR (low R_1), as it is illustrated in Fig. 11 for $R_1 = [10^{-3}, 0]$. As $2^\alpha R_1$ is larger than one, the benefit predicted from the erristor framework vanishes, whereas a more accurate analysis based on actual probabilities would still yield a gain. If the receive energy is taking into account, however, it may not be worthwhile to spend energy on listening to remote transmitters, since the incremental gain may be comparatively small. Therefore, we only consider transmitters with erristances smaller than one, and we limit m to 5. □

Finally, to demonstrate the applicability of the erristor concept to random networks, let us consider line networks with uniformly random node distribution.

Example 7. Random Line Networks: For long line networks with uniformly random distribution, the internode distances D_k are exponentially distributed and independent identically distributed (i.i.d.), and without loss of generality, we assume unit mean. For unit transmit power $(P = 1)$, the erristance equals D_k^α, and its pdf is

$$p_R(r) = \exp\left(-r^{\frac{1}{\alpha}}\right) \frac{r^{\frac{1}{\alpha}}}{\alpha r}. \qquad (28)$$

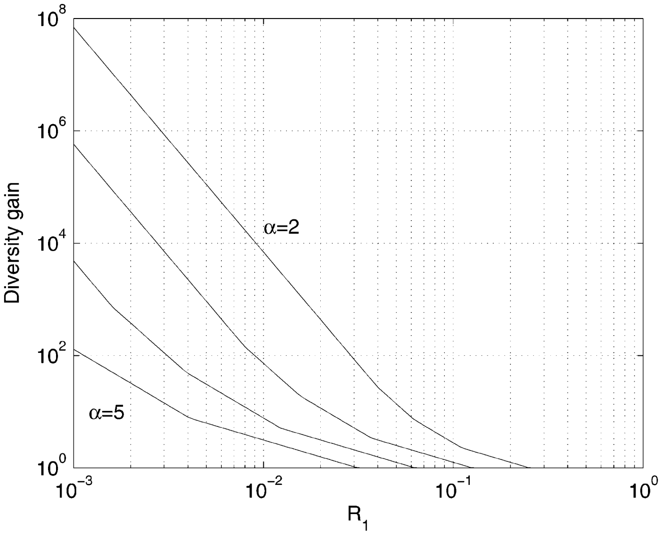


Fig. 11. Diversity gain from listening to m transmissions as a function of the single-transmission erristance R_1 for $\alpha = 2, 3, 4, 5$. $m = \min\{5, \lfloor R_1^{-1/\alpha} \rfloor\}$.

The mean of R is the αth moment of an exponential random variable (RV), so we have $\mathbb{E}[R] = \Gamma(\alpha+1)$ or, if α is an integer, $\mathbb{E}[R] = \alpha!$. For $P \neq 1$, this is to be divided by P.

For a two-neighbor cooperative scheme $(m = 2)$, the expected per-hop erristance is $\bar{R}_2 = \mathbb{E}[D_1^\alpha (D_1 + D_2)^\alpha]$, and in general, with transmit power P, we have

$$\bar{R}_m = \mathbb{E}\left[\prod_{j=1}^{m} R_j\right] = \mathbb{E}\left[\prod_{j=1}^{m} \frac{1}{P} \left(\sum_{k=1}^{j} D_k\right)^\alpha\right]. \qquad (29)$$

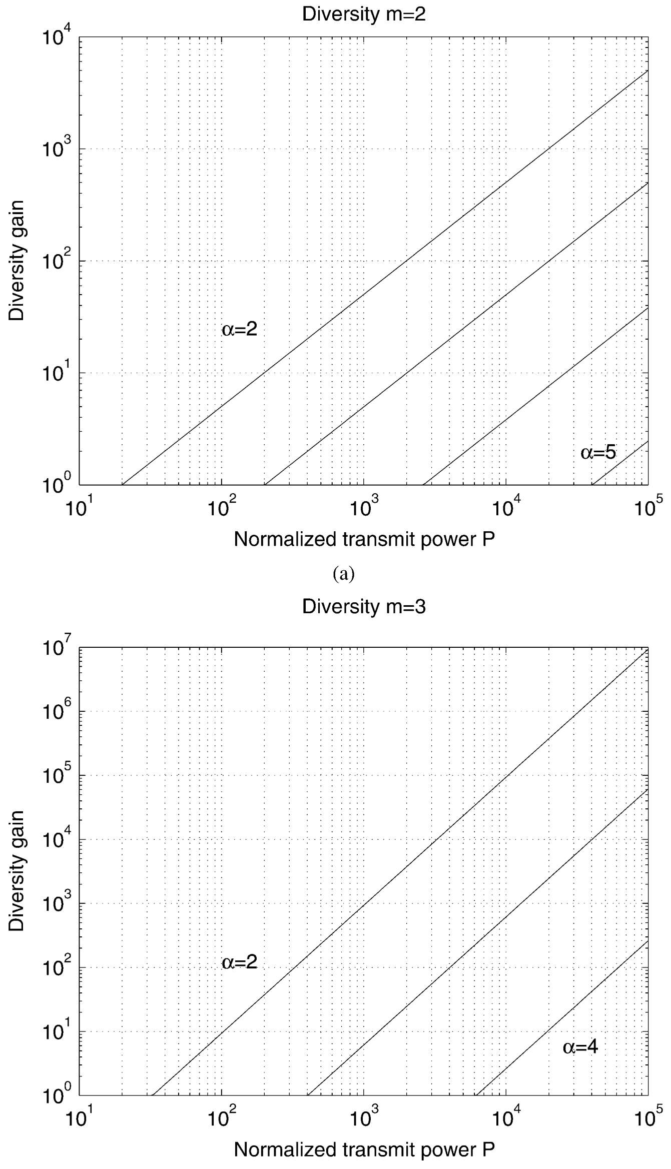
Since D_k is i.i.d., this can be simplified to sums and products of higher moments of D_k, which were determined above. Fig. 12 shows the gain $\bar{R}_1/\bar{R}_m$ for $m = 2$ and $m = 3$. The gain increases with P^{m-1}, and it is seen that the higher the path loss exponent, the higher the SNR has to be in order to get a benefit.

The minimum P necessary to see a benefit can be derived analytically. For $\alpha = m = 2$, e.g., we find $(\mathbb{E}[D^4] + 2\mathbb{E}[D^3]\mathbb{E}[D] + \mathbb{E}[D^2]^2)/P^2 < \mathbb{E}[D^2]/P$, or $40/P^2 < 2/P$, which holds for $P > 20$, in accordance with Fig. 12(a). □

V. Concluding Remarks

The erristor formalism permits the mapping of unwieldy probability expressions into a simple circuit-like framework, which greatly simplifies the analysis and design of transmit schemes that are based on selection combining with time diversity, path diversity, or a combination thereof. The wireless multicast advantage is incorporated by implicit erristors. The erristor circuit is topologically equivalent to the network graph and can, therefore, be drawn in a straightforward manner. Resource (re)allocation problems can effortlessly be solved by simple arithmetic, which makes a real-time implementation possible. A series of examples demonstrated not only the capabilities of this analysis and design tool, but also the huge gain that is possible from diversity-based communication strategies, in particular when high reliabilities are required. With more complex combining techniques, such as maximum ratio combining, this gain could be improved even further.

[6] This name is borrowed from analog RLC filter design.



(a)

(b)

Fig. 12. Diversity gain for random networks as a function of the normalized transmit power P for $\alpha = 2, 3, 4, 5$. (a) $m = 2$. (b) $m = 3$.

Further, the formalism is useful for educational purposes, since the multiplication property of parallel erristors impressively demonstrates the benefits of diversity schemes, and the series connection shows how the noise and, in turn, the packet loss probability accumulate over multiple hops.

The simple multinomial expressions for end-to-end erristances also lend themselves to a sensitivity analysis, i.e., the determination of the sensitivity of the total erristance to the individual erristances $\partial R_{\text{tot}}/\partial R_i$. This permits the allocation of power where it has the most impact.

Erristor circuits are relatives of both electronic circuits and signal flow graphs (SFGs), but some important differences should be pointed out. The analogy to electronic resistive circuits only holds for serial elements; the multiplication of parallel erristors does not have a counterpart in electric circuits. Further, the erristor diagram is *directed*, it requires information about the direction of the transmission, i.e., who is transmitting and who is receiving. To make this clear, an arrow could be used as in SFGs to indicate direction. This would also add timing information to the diagram, since it would indicate a natural sequence of the transmissions.

Compared with SFGs, the way series and parallel branches are treated is different. In an SFG, transmittances of serially connected branches are multiplied and parallel branches result in addition, whereas in the erristor diagram, serial branches are added and parallel branches are multiplied. If instead of NSRs, SNRs were used, the multiplication of serial branches would hold for both graphs, but k parallel branches would have to be treated like parallel resistors, yielding the harmonic mean divided by k. Further, the topology of the SNR-based circuit would no longer be identical to the original ad hoc network.

Appendix
Proof of Theorem 2

Proof: In the inequality [see (11)]

$$1 - \prod_{i=1}^{n}(1 - e^{-x_i}) \geq e^{-\prod_{i=1}^{n} x_i} \tag{30}$$

it is easy to see that equality holds (both sides are equal to 1) if one of the x_i is zero. It remains to show that $f_n(\,\cdot\,) : \mathbb{R}^n \to \mathbb{R}$

$$f_n(x_1, x_2, \ldots, x_n) = 1 - \prod_{i=1}^{n}(1 - e^{-x_i}) - e^{-\prod_{i=1}^{n} x_i} \tag{31}$$

is positive if all x_i are positive. We note that f goes to zero if $\forall i, x_i \to \infty$. Hence, f is positive for positive x_i if its partial derivatives $\partial f/\partial x_i$ are positive at zero and have at most one zero for positive x_i. Since the function is symmetric in all x_i, it is sufficient to consider only one partial derivative. An inductive technique is employed, discussing the case $n = 2$ first.

Consider

$$g_2(x) := f_2(x, y) = e^{-x} + e^{-y} - e^{-x-y} - e^{-xy} \tag{32}$$

for a fixed $y \geq 0$. For $y = 0, g_2(x) \equiv 0$, and for $y > 0$, we note that $g(0) = 0$ and $\lim_{x\to\infty} g(x) = e^{-y} > 0$. Since

$$g'(x) = \frac{dg}{dx} = e^{-x}(-1 + e^{-y} + ye^{x(1-y)}) \tag{33}$$

there exists a single local extremum $\bar{x}$ for $y > 0$ at

$$\bar{x} = \frac{1}{y-1} \ln\left(\frac{y}{1 - e^{-y}}\right). \tag{34}$$

As $y > (1 - e^{-y})$ for $y > 0$, we find $\bar{x} > 0$ for $y > 1$ and $\bar{x} < 0$ for $0 < y < 1$. For $y = 1$, no solution exists, and for $y = 0, g'(x) \equiv 0$. Since $g'(0) = -1 + e^{-y} + y > 0$, it is clear that the extremum is a maximum. So, we have $g(0) = 0$, and for $0 < y \leq 1, g(x)$ is monotonically increasing, whereas for $y > 1$, it is monotonically increasing up to $\bar{x}$, and then monotonically decreasing to $e^{-y} > 0$. Hence, $g(x) > 0$ for $x > 0$, and we have proven the theorem for $n = 2$.

Now, assuming it is true for $n - 1$, we prove the induction step.

For the general function $g_n(x) := f(x, x_2, \ldots, x_n)$ with fixed $x_i \geq 0$ for $i > 1$, we note that $g_n(0) = 0$ and

$$\lim_{x\to\infty} = 1 - \prod_{i=2}^{n}(1 - e^{-x_i}) \geq 0. \tag{35}$$

With $Q := \prod_{i=2}^{n} x_i$, we get

$$g_n'(x) = e^{-x}\left(-\prod_{i=2}^{n}(1 - e^{-x_i}) + Qe^{x(1-Q)}\right). \tag{36}$$

Evaluation at $x = 0$ yields

$$\begin{aligned} g'(0) &= Q - \prod_{i=2}^{n}(1 - e^{-x_i}) \\ &\geq 1 - e^{-Q} - \prod_{i=2}^{n}(1 - e^{-x_i}) \\ &= f_{n-1}(x_2, x_3, \ldots, x_n) \\ &\geq 0 \end{aligned} \tag{37}$$

where we have made use of the induction. Again, equality holds for $Q = 0$ only. Solving $g_n'(\bar{x}) = 0$ yields the single extremum

$$\bar{x} = \frac{1}{Q-1}\ln\left(\frac{Q}{\prod_{i=2}^{n}(1 - e^{-x_i})}\right). \tag{38}$$

We already established in (37) that the numerator is greater than (or equal to) the denominator in the logarithm, so, analogously to (34), we find that $\bar{x} < 0$ for $0 < Q < 1$ and $\bar{x} > 0$ for $Q > 1$. Again, no solution for $Q = 1$ and $g'(x) \equiv 0$ for $Q = 0$. ∎

References

[1] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1333–1344, Aug. 1999.

[2] A. Ephremides, "Energy concerns in wireless networks," in *IEEE Wireless Commun.*, vol. 9, Aug. 2002, pp. 48–59.

[3] A. J. Goldsmith and S. B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," in *IEEE Wireless Commun.*, vol. 9, Aug. 2002, pp. 8–27.

[4] D. Tse and S. Hanly, "Multi-access fading channels: Part I: Polymatroid structure, optimal resource allocation and throughput capacities," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2796–2815, Nov. 1998.

[5] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1277–1294, Jun. 2002.

[6] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, to be published.

[7] J. A. Silvester and L. Kleinrock, "On the capacity of multihop slotted ALOHA networks with regular structure," *IEEE Trans. Commun.*, vol. COM-31, no. 8, pp. 974–982, Aug. 1983.

[8] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Trans. Commun.*, vol. COM-32, no. 3, pp. 246–257, Mar. 1984.

[9] L. Hu, "Topology control for multihop packet networks," *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1474–1481, 1993.

[10] J. L. Wang and J. A. Silvester, "Maximum number of independent paths and radio connectivity," *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1482–1493, Oct. 1993.

[11] M. Sanchez, P. Manzoni, and Z. Haas, "Determination of critical transmission range in ad-hoc networks," in *Proc. Multiaccess, Mobility, Teletraffic Wireless Commun. (MMT'99)*, Venice, Italy, Oct. 1999.

[12] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[13] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proc. IEEE INFOCOM*, Anchorage, AL, 2001.

[14] G. Németh, Z. R. Turányi, and A. Valkó, "Throughput of ideally routed wireless ad hoc networks," *ACM Mobile Comput. Commun. Rev.*, vol. 5, no. 4, pp. 40–46, 2001.

[15] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy–latency–density design space," *IEEE Trans. Mobile Comput.*, vol. 1, no. 1, pp. 70–80, 2002.

[16] T. J. Shepard, "A channel access scheme for large dense packet radio networks," presented at the ACM SIGCOMM, Stanford, CA, Aug. 1996, [Online]. Available: http://www.acm.org/sigcomm/sigcomm96/papers/shepard.ps.

[17] E. S. Sousa and J. A. Silvester, "Optimum transmission ranges in a direct-sequence spread-spectrum multihop packet radio network," *IEEE J. Sel. Areas Commun*, vol. 8, no. 5, pp. 762–771, Jun. 1990.

[18] D. A. Maltz, J. Broch, and D. B. Johnson, "Lessons from a full-scale multihop wireless ad hoc network testbed," *IEEE Pers. Commun.*, vol. 8, no. 1, pp. 8–15, Feb. 2001.

[19] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. (2002) An empirical study of epidemic algorithms in large scale multihop wireless networks. Intel Res. Rep. IRB-TR-02-003. [Online]. Available: www.intel-research.net/Publications/Berkeley/050 220 021 703_19.pdf

[20] T. S. Rappaport, *Wireless Communications—Principles and Practice*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[21] M. Zorzi and S. Pupolin, "Optimum transmission ranges in multihop packet radio networks in the presence of fading," *IEEE Trans. Commun.*, vol. 43, no. 7, pp. 2201–2205, Jul. 1995.

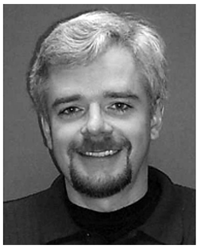
[22] M. Haenggi, "Probabilistic analysis of a simple MAC scheme for ad hoc wireless networks," in *IEEE CAS Workshop Wireless Commun. Netw.*, Pasadena, CA, Sep. 2002.

[23] X. Liu and M. Haenggi, "Throughput bounds and energy consumption of mobile multihop networks," in *Proc. IEEE Veh. Technol. Conf. (VTC'04 Fall)*, Los Angeles, CA, Sep. 2004.

[24] J. N. Laneman and G. W. Wornell, "Energy-efficient antenna sharing and relaying for wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC'00)*, Chicago, IL, 2000, pp. 7–12.

[25] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000, pp. 585–594.

[26] P. Gupta and P. R. Kumar, "Toward an information theory of large networks: An achievable rate region," *IEEE Trans. Inf. Theory*, vol. 49, no. 8, pp. 1877–1894, Aug. 2003.



Martin Haenggi (S'96–M'99–SM'04) received the Dipl. Ing. (M.Sc.) degree in electrical engineering, the Dipl. NDS ETH (post-diploma) degree in information technology, and the Ph.D. degree from the Swiss Federal Institute of Technology (ETHZ), Zurich, Switzerland, in 1995, 1996, and 1999, respectively. His Ph.D. dissertation was on the analysis, design, and optimization of cellular neural networks.

In 1995, he joined the Signal and Information Processing Laboratory, ETHZ, as a Teaching and Research Assistant. After a postdoctoral year at the Electronics Research Laboratory, University of California, Berkeley, he joined the faculty of the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, as an Assistant Professor in January 2001. His scientific interests include networking and wireless communications, with an emphasis on ad hoc and sensor networks.

Dr. Haenggi was awarded the ETH Medal for both his M.Sc. thesis and Ph.D. dissertation. He received the 1995 Willi Studer Award at ETH.

Appendix D: PI Presentations

Electronic Copies of the PI Presentations accompany the Final Report

Real-time Reconfiguration of Networked Embedded Systems

OBJECTIVE:

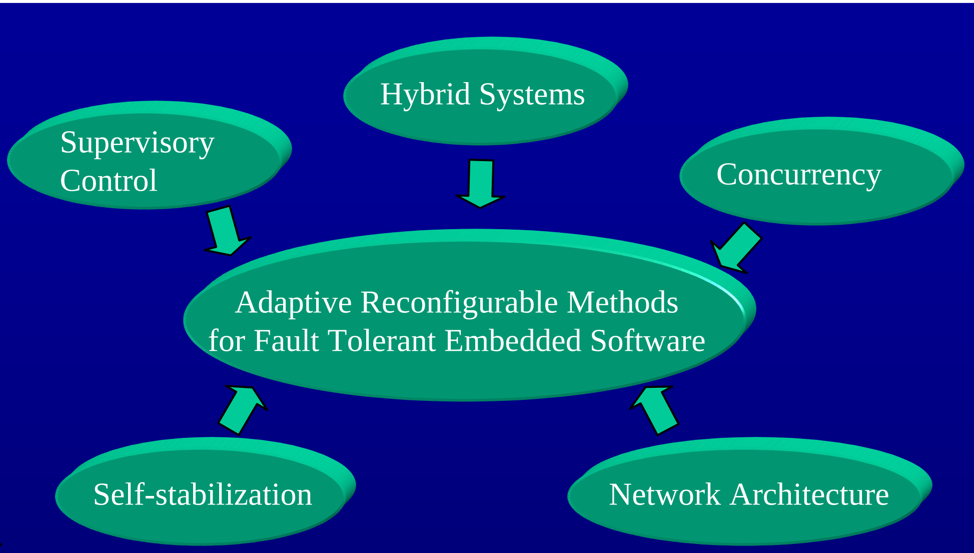
“Efficient and adaptable dynamic software reconfiguration and fault-tolerant synchronization and security in fine-grained networks of embedded systems that interact tightly with the physical world”

UNIVERSITY OF NOTRE DAME

TEAM COMPOSITION:

- **P.J. Antsaklis, Dept. of Electrical Engineering**
- **M.D. Lemmon, Dept. of Electrical Engineering**
- **M. Haenggi, Dept. of Electrical Engineering**

REAL-TIME RECONFIGURATION OF NETWORKED EMBEDDED SYSTEMS



NEW IDEAS

•Hybrid Systems for the formal analysis of self-stabilizing algorithms.

•Supervisory Control: a formal basis for S/W design

•Advanced Concurrency Models for scalable analysis and design methods

•Self-stabilizing Algorithms for fault-tolerance and security

•Advanced Architectures for wireless real-time networks

IMPACT

•Automated Software Reconfiguration in real-time networks of embedded systems

•Scalable and Dependable Methods for real-time software analysis and synthesis

•Dependable Data and Time Synchronization across highly dynamic networks of embedded systems

SCHEDULE

Year 1
•Models and Algorithms for Adaptive Reconfiguration
•Analysis of Synchronization and Security Algorithms

Year 2
• Development of Algorithms and Software

Year 3
• Implementation and Integration

Project Issues

- **<u>TASK 1</u>**

  How do we adaptively reconfigure embedded node's O/S kernel?

  - **external changes in the real-world**
  - **changes in network links and topology**
  - **scalability (with respect to network size)**

- **<u>TASK 2</u>**

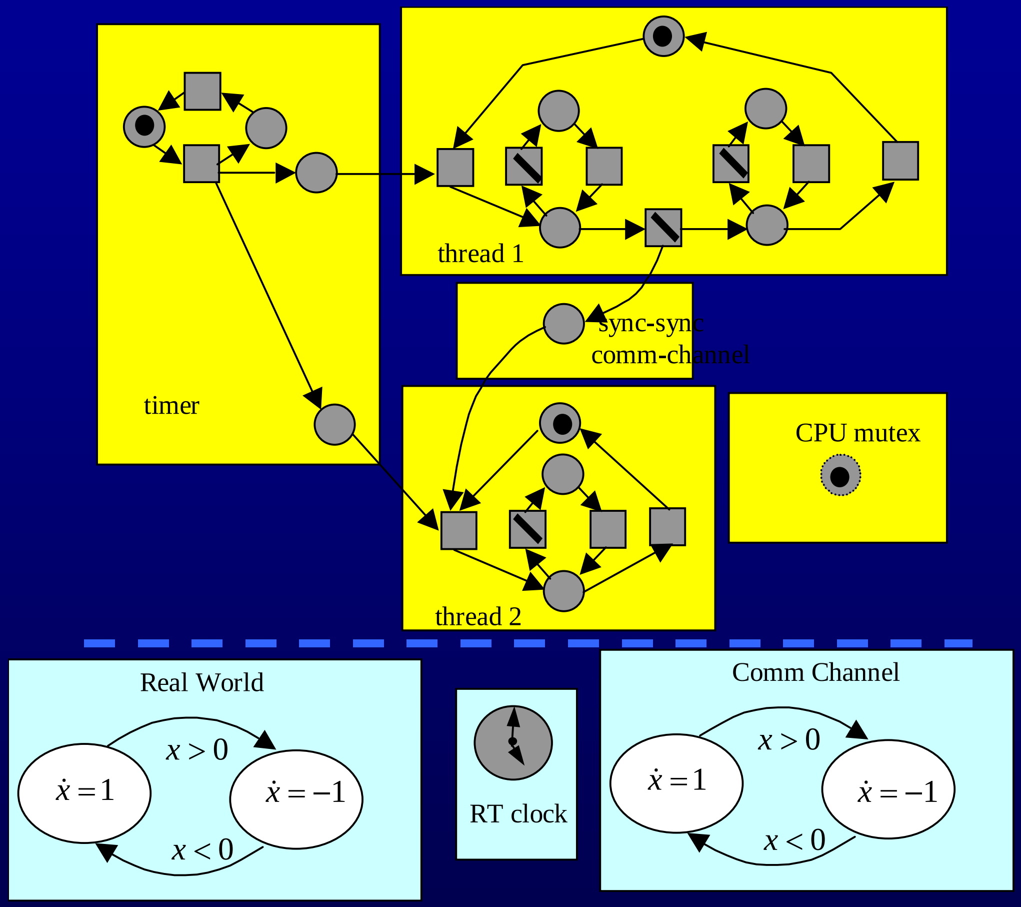
  How do we synchronize process execution across network nodes in the absence of a global clock?

  - **fault-tolerance through self-stabilizing algorithms**
  - **generalized clock models**

    (synchronizing periodic tasks in the face of real-world disturbances)
  - **highly decentralized**
  - **Network Security (two-faced intruders)**

Adaptive Reconfiguration (Formal Models)

- O/S Kernel modeled by hybrid Petri nets
- Real-world modeled by hybrid automata



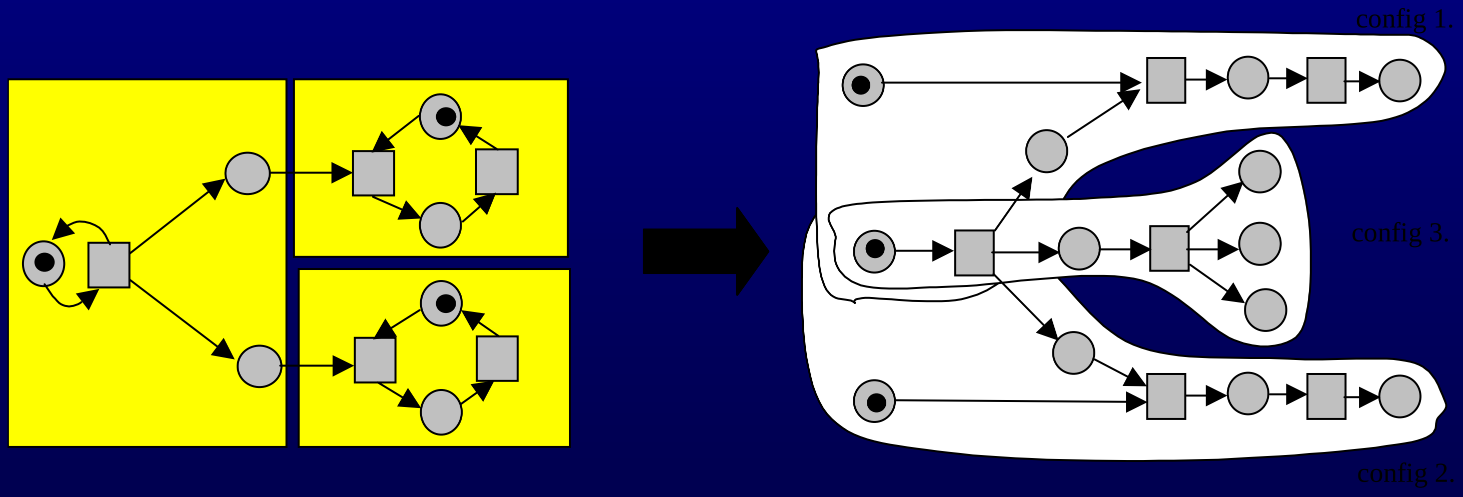
Hybrid models capture inherent timing and interaction of processor tasks with network and real-world

Partial Order Analysis of System Models

- Transformation of system models into an event structure maps original net system into an acyclic occurrence net that preserves event order, preserves choice, and enumerates reachable states in a compact manner

- Extract a set of “configurations” that act as a “basis” for the behaviors that can be generated by the system

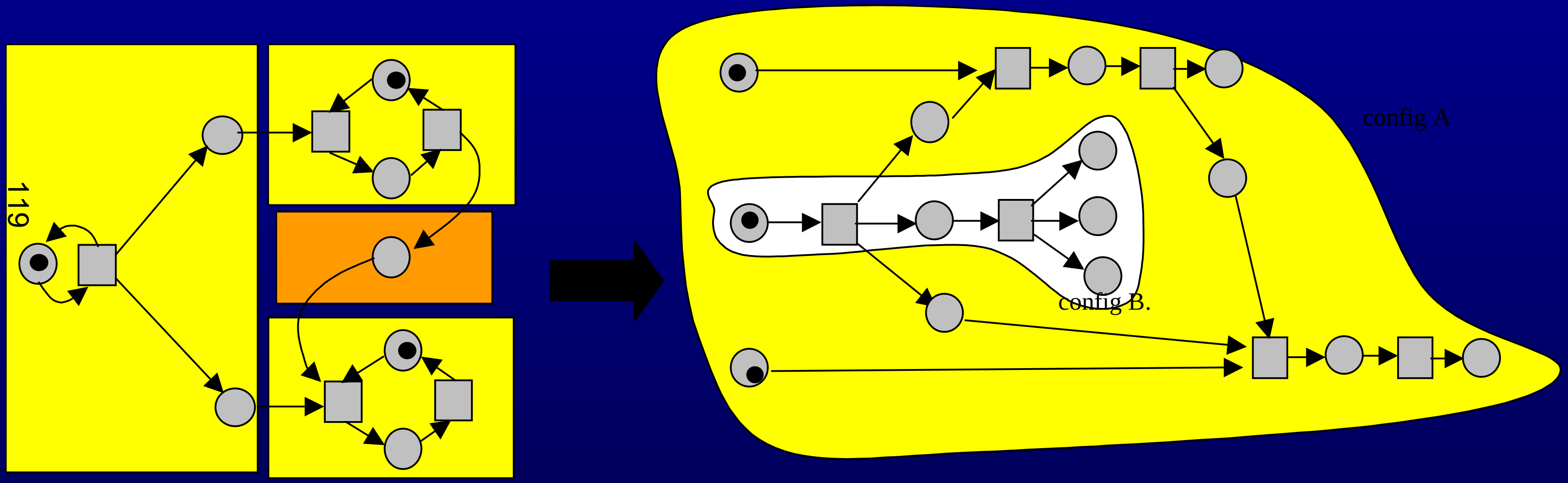
$$L(S) = L(C_1) \parallel L(C_2) \parallel L(C_3)$$



Supervisory Control of Periodic Tasks

- LTL (linear temporal logic) specifications on task executions can be enforced through a “supervisory” strategy that disables undesirable events

$$L(S \mid \alpha) = \alpha\left[L(C_1) \,\|\, L(C_2)\right] \| \, L(C_3) = L(C_A) \,\|\, L(C_3)$$

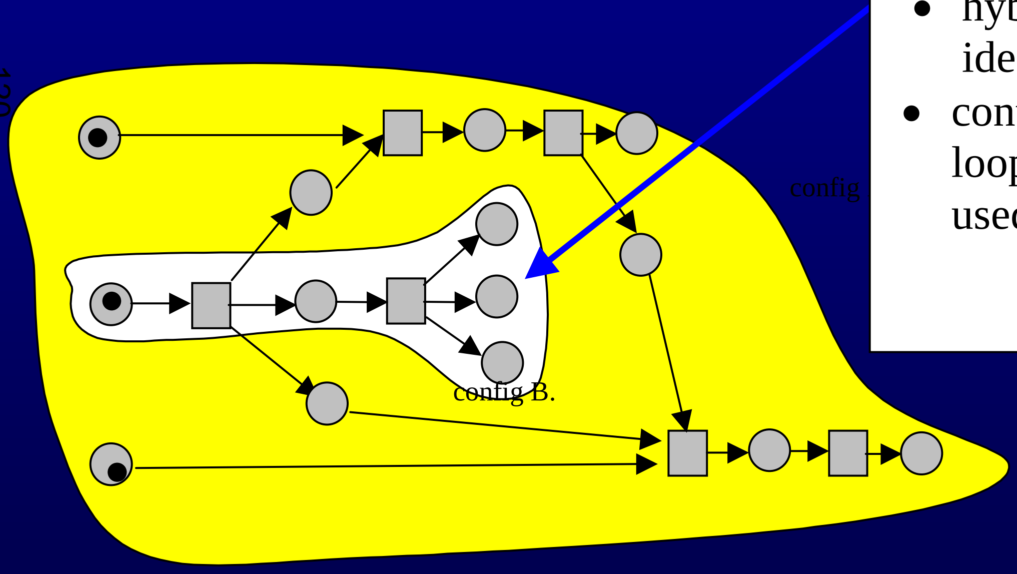


Partial Order Analyses provide a means of automatically synthesizing supervisors that restrict system’s aggregate behavior.

We intend to use this approach to develop an embedded O/S kernel that can automatically upgrade itself and reconfigure to changes in external world

Hybrid Analysis of Real-time Systems

- Event structures are labeled with guards that condition events on satisfaction of guard predicates that are functions of continuous-variables.
- Use of hybrid control system methodologies to disable undesirable configurations



Timer Overflow:

- hybrid system analyses can be used to identify this problem
- control methodologies such as closed loop adjustment of task periods can be used to disable unsafe configuration

Adaptive Reconfiguration Summary

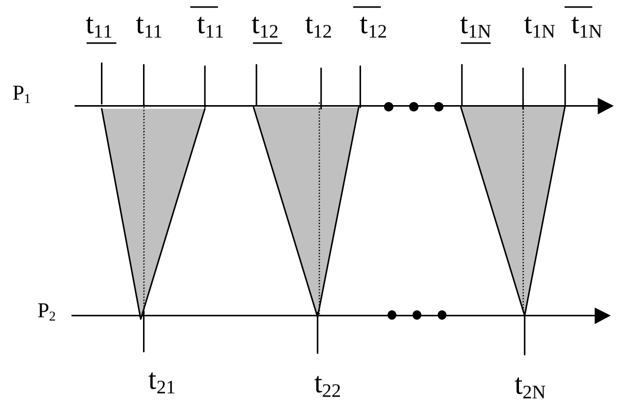
- Formal hybrid network models of node O/S, communication channel, and external environment
- Use of partial order method (unfolding) to identify a basis of *configurations*
- Automatic synthesis of supervisory policies that enforce LTL specifications on system behavior
- Automatic synthesis of feedback control laws that disable unsafe system executions due to hybrid phenomena such as timer-overflows

Develop an adaptive O/S kernel that can reconfigure in the face of changes in network structure as well as the external world

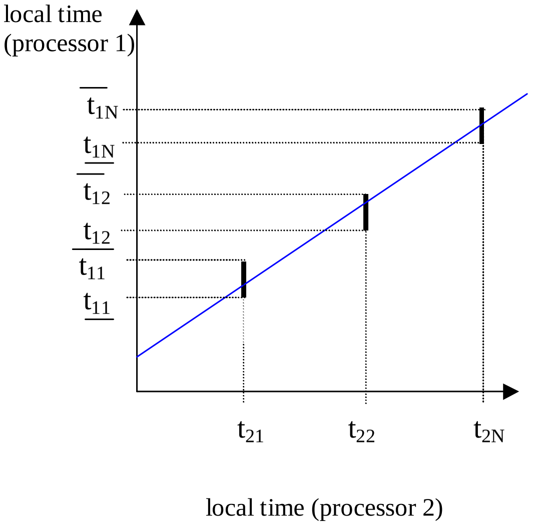
The use of partial order methods provides an inherent compositionality of supervisory goals that allow us to gracefully scale our methods to large scale networks.

Use of hybrid system methodologies allow us to explicitly account for network's tight interaction with the external world

HYBRID SYSTEM ANALYSIS OF REAL-TIME SYNCHRONIZATION



space-time diagram showing message passing

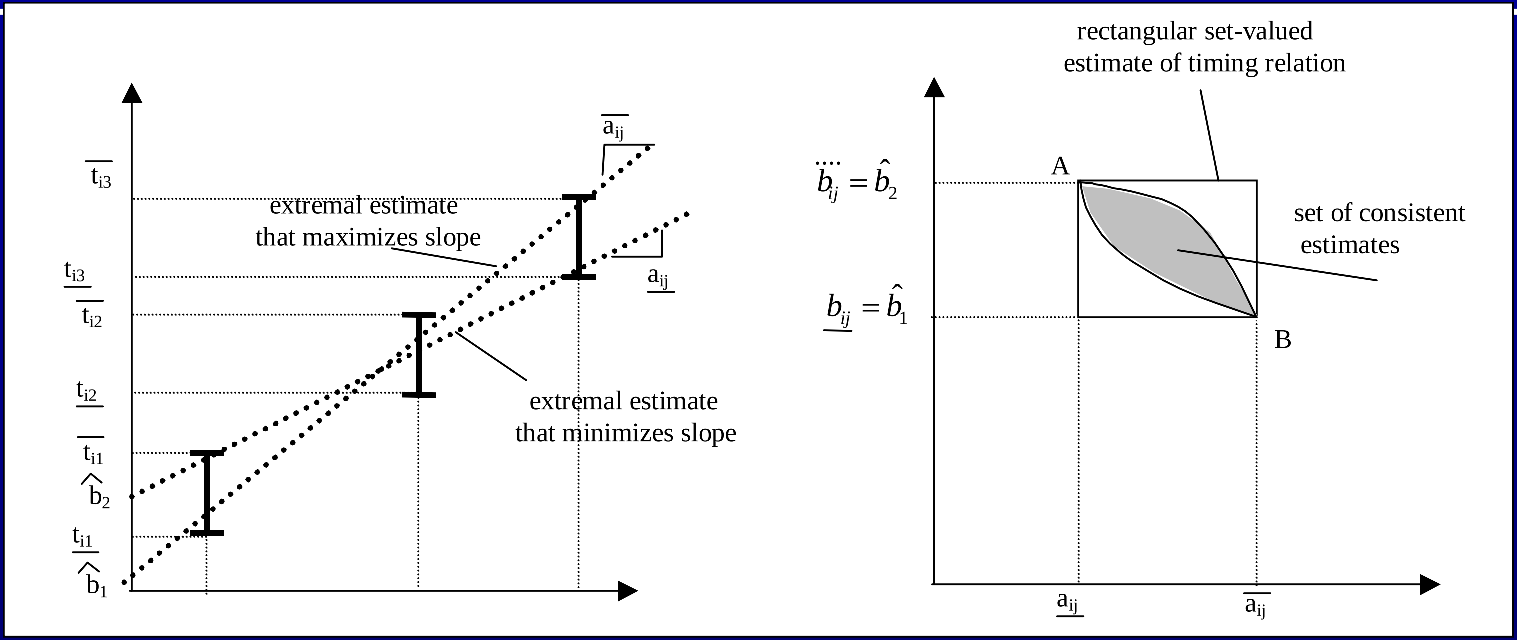


- Clock synchronization in the absence of a global clock
- Time-stamped UDP packets

Model-based Synchronization

- Identify drift/offset of neighboring clocks from passed messages.
- These estimated parameters form a "model" of neighboring processors that can be used to authenticate messages and correct for Byzantine faults.

SET-VALUED CLOCK MODELS, FAULT DETECTION, and ACCOMMODATION



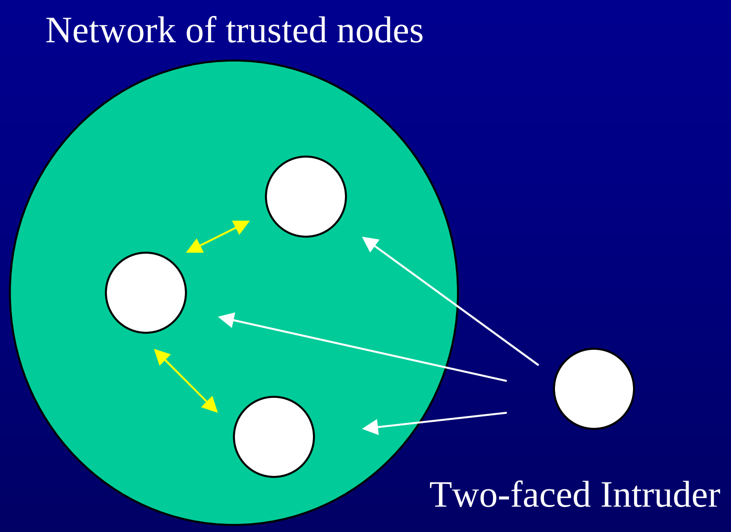
Set-valued Clock Models

- Set-valued estimate of clock parameters obtained from LP
- Set valued estimate is **optimal** in sense of minimizing area of rectangular shaped set estimate.

Fault Detection and Accommodation

- Faults detected between two clocks if subsequent time-stamps are inconsistent with identified model
- Analytic prediction of "stale" models that can be used to schedule resynchronization

TWO-FACED CLOCKS and NETWORK SECURITY



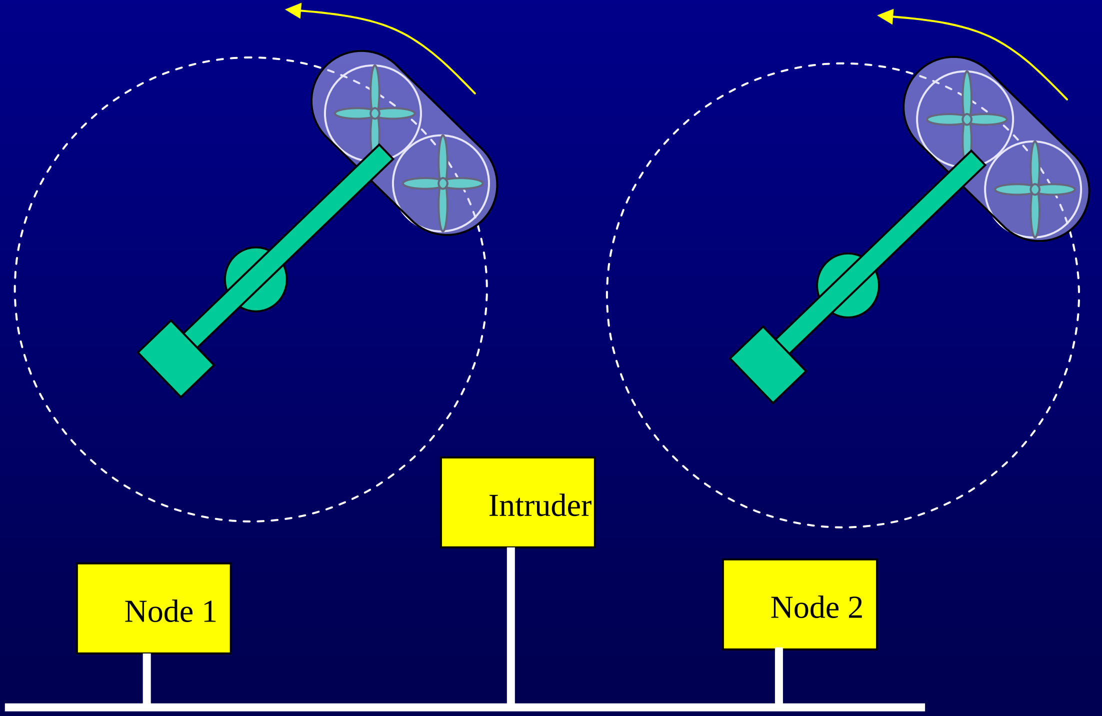
•Two-faced Clock:
A node that provides one time-stamp to one node and a different time-stamp to another node

•Authenticated Time-stamps are “signed” by a signature that is unique to the sending node

•Model-based time-stamps are authenticated messages since the clock models are unique to a specified pair of nodes

•Detection of Two-faced Clocks takes advantage of the fact that set-valued clock models are transitive. So a pair of trusted nodes can compare the times an intruder transmits to them in order to detect two-faced clocks.

HARDWARE IMPLEMENTATION



- Experimental setup to test performance of model-based sync algorithm
- Two 3DOF helicopters under the control of separate computers
- One intruder node
- Systems synchronize on the “travel” (position) of each helicopter and perform coordinated maneuvers

Synchronization Summary

•Model-based synchronization method constructs dynamic models of neighbors clocks.

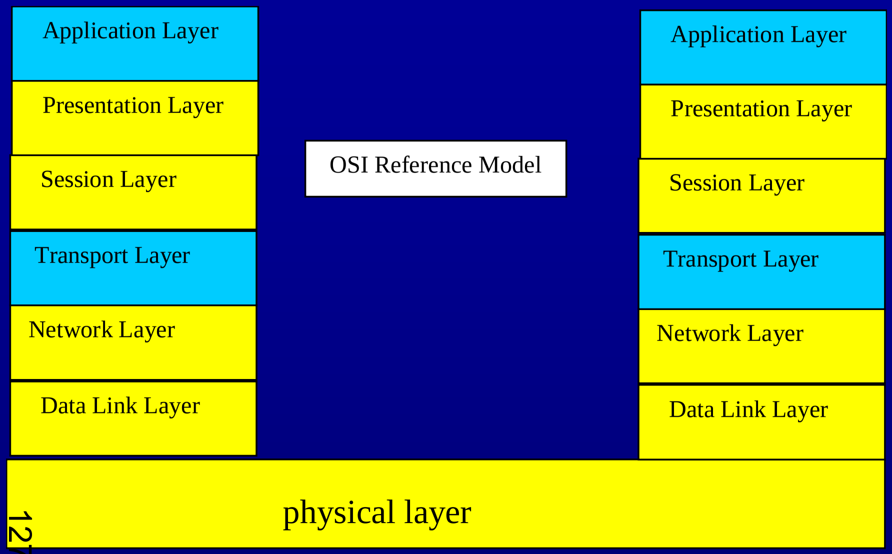
•Generalize notion of clock to any periodic process

•Optimal set-valued estimates of clock parameters provide a convenient way of detecting node faults

•Highly decentralized protocol similar to Dolev's synchronization algorithm

•Appears to be an *authenticated* algorithm so:
- no limitation on number of faulty processors
- may be useful in addressing Byzantine faults

Group II Interface



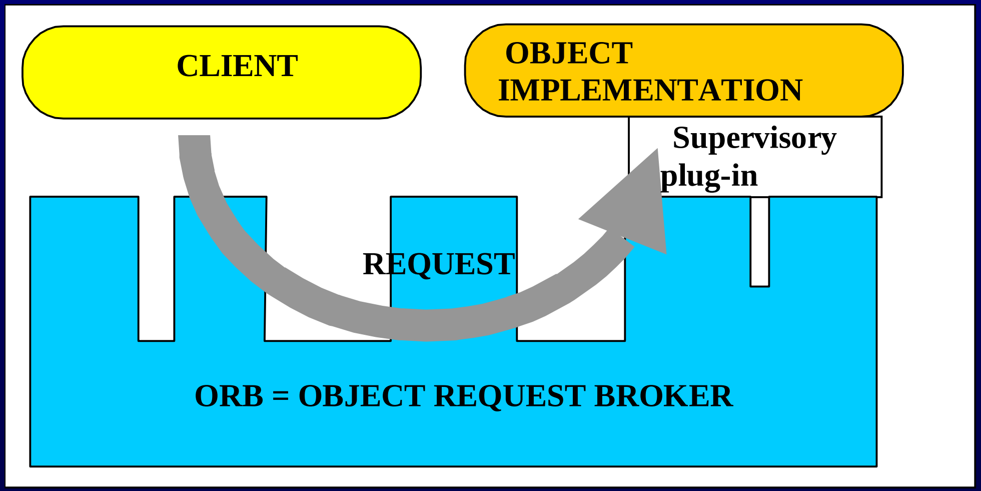
OSI REFERENCE MODEL

- ND services
  - synchronization (transport)
  - Reconfigurable O/S kernel (application layer)

•Group II services
- physical, data-link, network
- session, presentation

CORBA

- Reconfiguration of distributed objects through *supervisory plug-ins*



Deliverables and Milestones

- Year 1:
  - development of models and analysis of algorithms for adaptive reconfiguration
  - analysis of synchronization and security algorithms. Software development
- Year 2:
  - Development of algorithms and S/W for adaptive reconfig.
  - Development of algorithms and S/W for synchronization and security
  - Group II coordination/integration
- Year 3:
  - Integration of the adaptive configuration, synchronization and security algorithms.
  - Group II testbed full integration

Deliverables and Milestones

Fiscal Year 2001

- Deliverables:
  - Preliminary adaptive O/S kernel demonstration on 7-node QNX network
  - Hardware demonstration of synchronization algorithm 3-DOF helicopter demonstration

- Exploratory Tasks:
  - alternative verification methods (Ugo Buy, UIC)
  - sensor network (Motorola 68HC12 target processor)
  - Bluetooth technology survey
  - video-streaming
  - scaling studies

Networks of Embedded Systems
WORKING GROUP MEETING

September 11-13, 2001

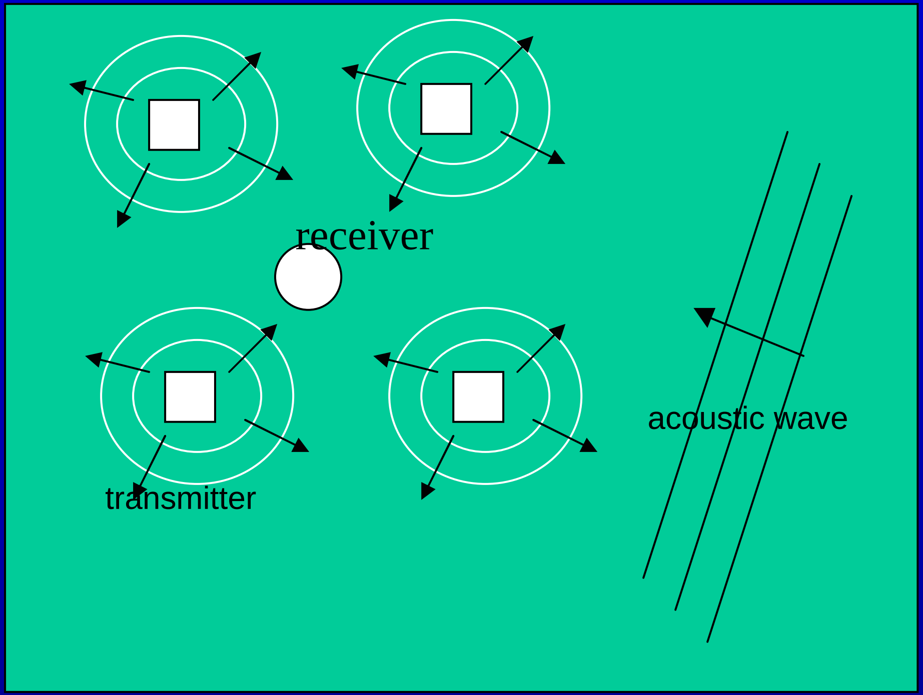
Huntington Beach, California

Project Overview

- Analysis of Composed Middleware Services (off-line)
  - Develop formal models (timed PN) of TinyOS components
  - Partial order analysis of composed models
  - Verification of composed service properties
    deadlock, liveness, fairness..
  - Synthesis of re-configuration supervisors and controllers enforcing composed service properties
- Fault-tolerant Time Synchronization Service (on-line)
  - Set-valued estimates of neighbor clocks (drift and offset)
  - Service transforms time-stamps to local node’s clock reference
  - Time scope of time-stamped messages and data
  - Faulty processors detected by inconsistent time-stamps
  - Fault accommodation (re-synchronization) to ensure data integrity
- Development Plan
  - Initial development on QNX network
  - Port to TinyOS

Model Problem

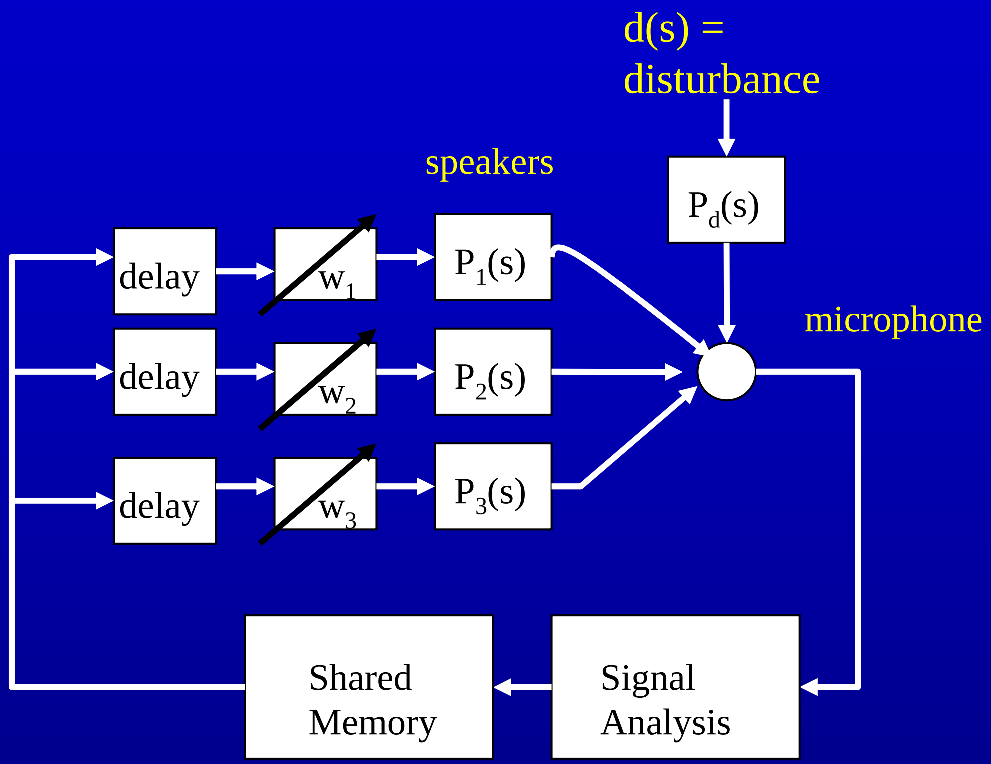
- Provide concrete framework upon which to identify required NEST middleware services.
- Adaptive Noise Cancellation (ANC)
- NEST protects shared resource that is threatened by narrow-band acoustic disturbance. Transmitters coordinate their actions to null disturbance at receiver
- NEST attributes
  - Coordination of distributed nodes
  - Closed-loop control
  - Real-time performance specs
  - Inherent security problem
  - Dynamic network environment



- ANC problem is well understood and lends itself to distributed solutions.
- Timing and localization requirements are consistent with micro-controllers
- Focus on middleware services rather than the “application” software

Model Problem (2)

- Receiver observes disturbance and transmitter actions
- Observed signals analyzed and time-stamped keys to data broadcast.
  (shared memory a la LIME)
- Transmitters access data through keys.
- Transmitter remains inactive if requested data is too old
- Active transmitters adjust oscillator magnitude and phase using an LMS algorithm



Other model problems are being considered
- atmospheric control, robotic colonies,
  pursuit evader problems

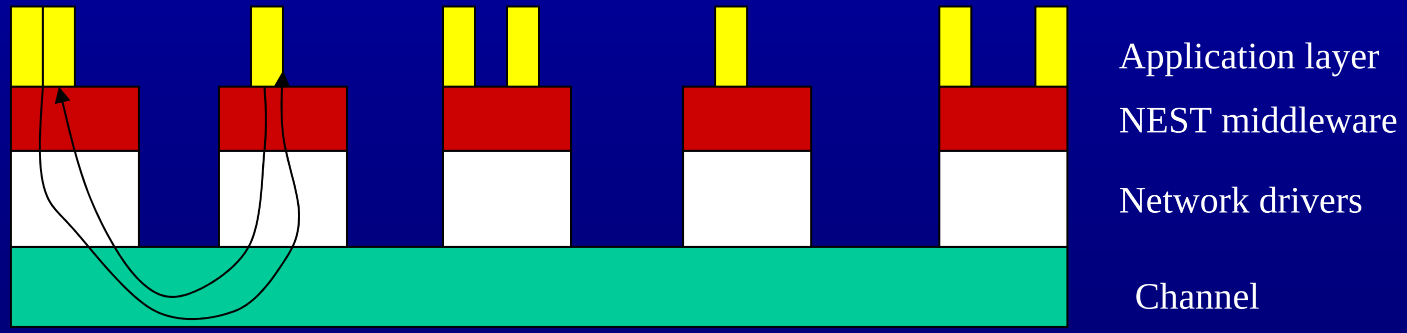
ANC Middleware Services

- Data-flow paradigm (consumers pull data from network)
- NEST Middleware services provide

  - message transformation
  - network protocols
  - high level API for application layer

  - time synchronization
  - localization service
  - shared memory management
  - security services
  - group maintenance

- Since NEST will use a variety of middleware services, we'll need to analyze service interaction with respect to system properties



Fault-Tolerant Time Synchronization

- Synchronization service identifies local clock models and allocates local node addresses
- Time transformation Service automatically transforms received message time-stamps to local clock
- Garbage Collection Service discards shared data with expired time tags
- Faulty node detection and accommodation service

			Embedded Software Focus Areas		
			Middleware Services	Configurations	
				On-line	Off-line
Technology Challenges	*Res. Mgmt*	CPU			
		Memory			
		Network			
		Power			
	Fault Tolerance		■	■	
	Time Synchronization		■	■	
	Heterogeneous Processing				
	Safety Criticality				

Time Sync Requirements

- Platform
  - 1-5 k-bytes RAM, 30-60 k-bytes EEPROM acoustic speaker/microphone expansion boards
  - Release Restrictions – NONE
  - Mobile Mote Platform
- TinyOS Components
  - Background task scheduler
- Additional NEST Services
  - Shared memory services
  - Multi-hop routing service (?)
  - Localization service

Service Composition Tool

- Model Building Tool
  DESC file → PN model
- Model Analysis Tool
  deadlocks, liveness, fairness
  CPU, memory, power usage
- Supervisor Synthesis Tool
  synthesizes PN supervisor
  for composed service

			Embedded Software Focus Areas		
			Middleware Services	Configurations	
				On-line	Off-line
Technology Challenges	*Res. Mgmt*	CPU			■
		Memory			■
		Network			
		Power			■
	Fault Tolerance				
	Time Synchronization				
	Heterogeneous Processing				
	Safety Criticality				■

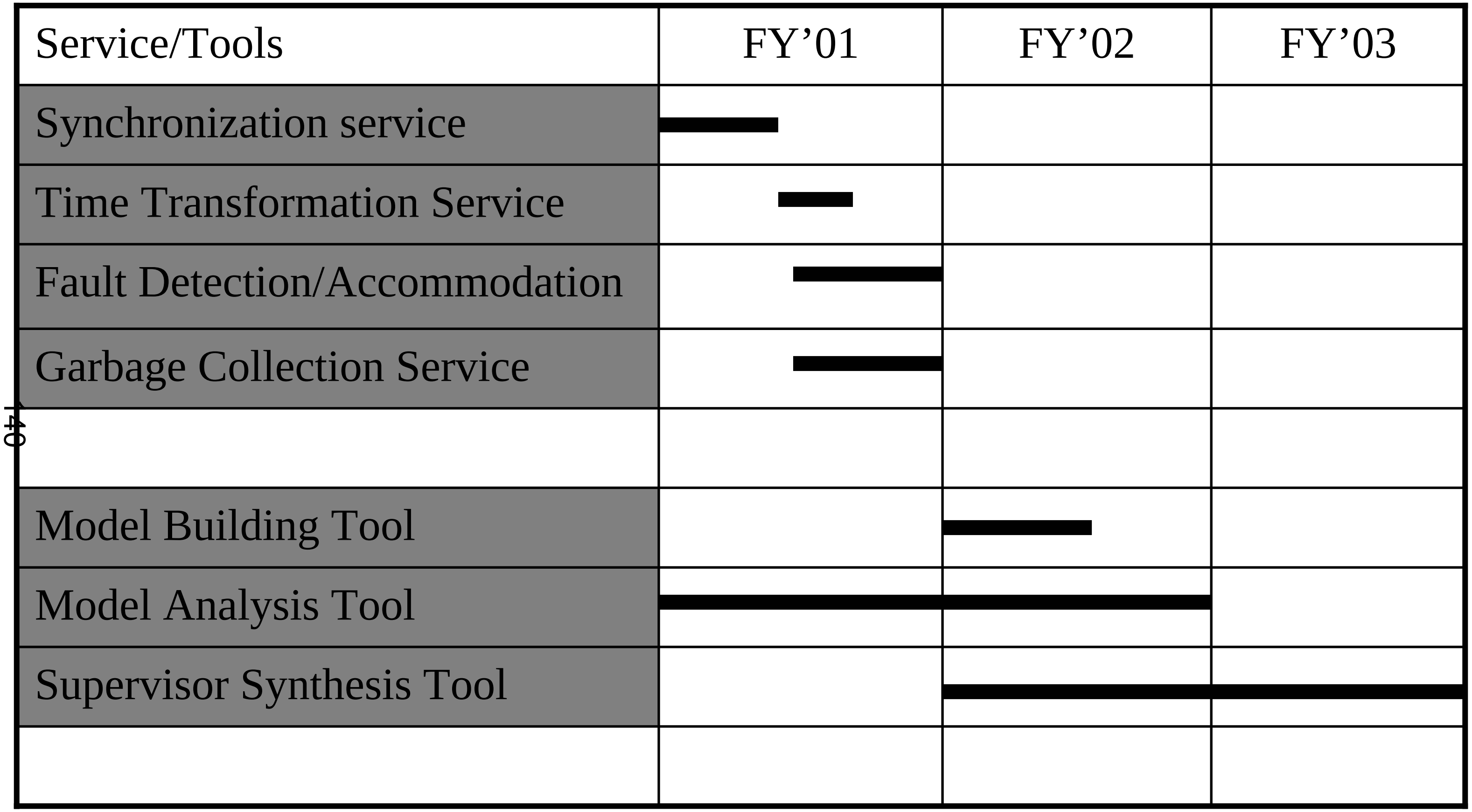
Service Tool Requirements

- DESC file syntax is insufficient to identify data and control flow between TinyOS components.
- This means Timed Petri Net models must be generated by hand which leads to obvious correctness issues
- A richer DESC file syntax should:
  - Declare choice and synchronization
  - Declare guard conditions on choices
  - Declare worst-case resource usage of spawned threads
  - Declare assumed continuous dynamical models for external world
- Hybrid models of external environment
  - to analyze the effect composed services have on mote CPU utilization, memory allocation, and power consumption, we need formal models of physical processes being controlled
  - Models should be provided by application designer.

OEP Requirements and Integration

- OEP Platform Requirements
  - More memory (at least)
    1-5 k-bytes variable memory
    30-60 k-bytes program memory
  - Expansion boards supporting ANC model problem
  - Platform mobility
- OEP Product Requirements
  - TinyOS development environment (okay)
  - Background task scheduler for TinyOS
  - Multi-hop routing component (?)
- Group 1 Products
  - Shared memory services
  - Localization service

Schedule


Service/Tools
FY’01
FY’02
FY’03
Synchronization service
Time Transformation Service
Fault Detection/Accommodation
Garbage Collection Service
Model Building Tool
Model Analysis Tool
Supervisor Synthesis Tool

Real-time Configuration of Networked Embedded Systems

University of Notre Dame

Contract No. F30602-01-2-0526
AO No: L546/00

Period: 5/2001 – 9/2004

Panos J. Antsaklis, Michael D. Lemmon, Martin Haenggi
Department of Electrical Engineering
PI Phone No: 574-631-5792, Fax 574-631-4393, antsaklis.1@nd.edu

DARPA Program Manager: Vijay Raghavan

Subcontractors and Collaborators

Senior Personnel

Panos J. Antsaklis, Michael D. Lemmon, Martin Haenggi

Electrical Engineering, University of Notre Dame

Project Consultant

Ugo Buy, Computer Science, University of Illinois-Chicago

Formal models and analysis of embedded system network –Berkeley motes and TinyOS

Graduate Students

Lucia Xia, Luis Montestruque, M. Brett McMickell

Qiang Ling, Ying Shang, Xiaowen Liu, Haihao Shi,

Marian Iordache, Hai Lin, Lei Fang,

Yashan Sun, Wenyi Zhang

Problem Description and Program Objective

A*daptive reconfiguration in fine grained* sensor-actuator-networks *of embedded systems* that interact tightly with the physical world so to maintain Quality of Service (QoS.)

Considering sensor-actuator-networked-systems (SANETS)-
vs sensor-only-networked-systems (SONETS)

Quality of Service Metrics

- Delays (max end to end)
- Packet dropout rates (probability of dropout)
- Node connectivity (number of node failures before connection is broken)

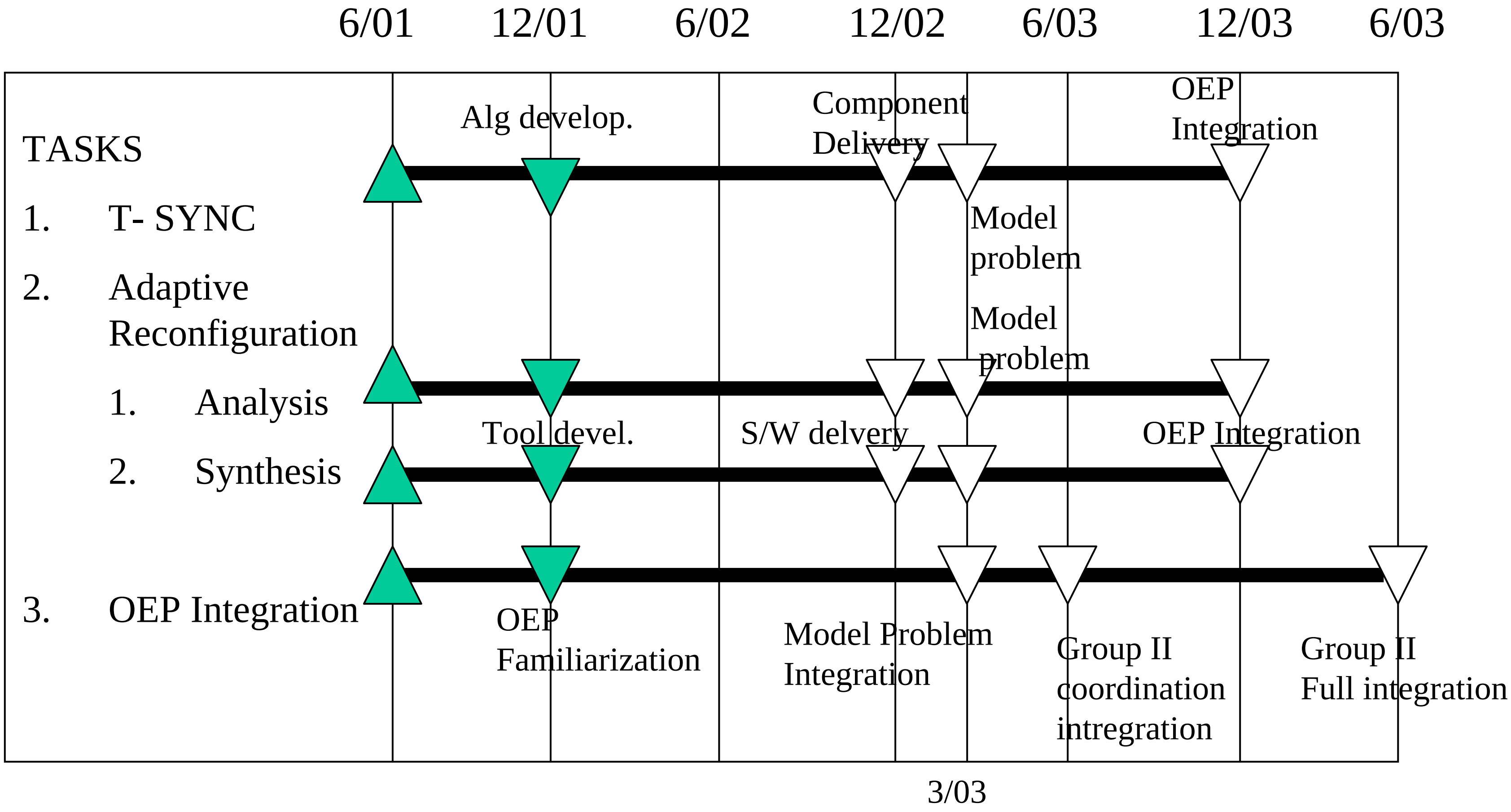
Contributions to NEST goals

- Projects addresses issues in SANETS and develops methods assuring system performance in the presence of NEST’s ad hoc networking environment.
- Primary goal is to ensure that middleware services do not degrade overall closed loop performance.

PROJECT TASK SUMMARY

- Time Synchronization
  develop time-sync algorithms for heterogeneous networks with significant clock drift
- Adaptive Reconfiguration of Sensor Actuator Networks (SANET)
  - QoS Analysis
    - Control system analysis provides bounds on SANET QoS (end-to-end delay, data dropout probabilities, connectivity)
    - Reachability Graph Analysis to estimate SANET QoS and identify “critical” transitions leading to QoS violation
  - Supervisor Synthesis
    - Synthesize S/W control structures to assure NEST middleware end-to-end delay and dropout rate
  - Link Reconfiguration
    - adaptively re-route messages in presence of link/node failures.
- OEP Integration
  - Model Problem (SANET) implemented with Berkeley OEP
  - Development of Time-Sync TinyOS components

SCHEDULE



6/01 12/01 6/02 12/02 6/03 12/03 6/03

TASKS

1. T- SYNC

2. Adaptive Reconfiguration

   1. Analysis

   2. Synthesis

3. OEP Integration

Alg develop.

Component Delivery

Model problem

OEP Integration

Model problem

Tool devel.

S/W delvery

OEP Integration

OEP Familiarization

Model Problem Integration

Group II coordination intregration

Group II Full integration

3/03

PROJECT STATUS

- No substantive changes in technical approach
  - examination of adaptive routing
- PROGRESS SINCE LAST MTG
  - T-Sync algorithm supports 100 Hz coordination frequency
  - Tools developed for analysis of NEST middleware in SANETS
  - Analysis and supervisor synthesis for 3-mote network
  - Familiarization with Berkeley OEP, setup lab facility supporting TinyOS component development
- 2 journal pubs (IEEE Trans Auto Control , IEEE Trans Robot/Auto)
  1 journal paper submitted, 1 more in preparation
  3 conference papers
- Milestones
  - all task milestones achieved for first meeting
  - no delayed milestones

OEP Participation

- Notre Dame is working with the Berkeley OEP
- Notre Dame’s role:
  - Development of model-based time-sync components
  - SANET model problem using Berkeley OEP
  - Analysis and Supervision of
    NEST Middleware for SANETs
- Technical Point of Contact
  David Culler (sourceforge)

PROJECT PLANS

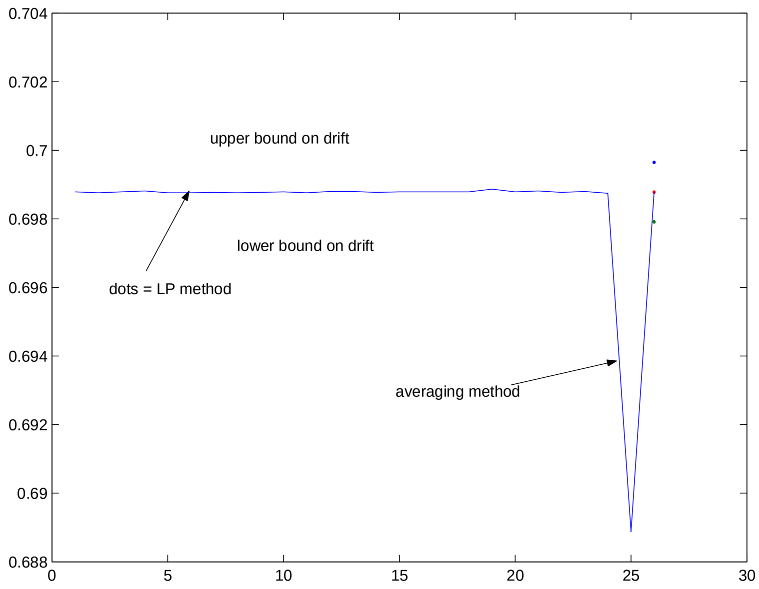
- Plan over next 6 months
  - Develop TinyOS T-sync component
  - Implement Model Problem (SANET) using Berkeley OEP
  - Develop Petri Net Models of SANET used in model problem
  - Analysis of NEST Middleware on model problem
- Performance Goals
  - Quantify QoS required by model problem
  - Estimate Middleware QoS level and experimentally verify

Time Synchronization Task (Lemmon)

- Purpose:
  ensure “synchronous” interaction of nodes
  - message passing, sensor sampling, actuator hold functions
- Problem:
  While t-sync methods exist for drift-free systems, SANETS are heterogeneous networks with significant clock drifts. We need to develop distributed fault-tolerant ways to synchronize clocks in the presence of such drift.
- Approach:
  Each node “estimates” relative clock drift and offset of its neighbors
  - Set-valued estimates that bound drift/offset errors
  - Traditional averaging methods
  - Middleware component transforms local times
- SUCCESS METRIC:
  Development of T-sync components that allow synchronous SANET operation consistent with coordination frequencies of 10-100 Hz.

Time Synchronization Status and Plans

- STATUS:
  - Development and analysis of model-based estimation method
  - Implementation on 6-node TCP/IP network (QNX)
  - Journal paper in preparation
- FINDINGS:
  - Drift accuracy = 1/4000, which implies that we can achieve 100 Hz synchronous coordination (5kHz, 16-bit clocks, light network traffic.
- FUTURE PLANS:
  - TinyOS component
  - Model Problem Implementation

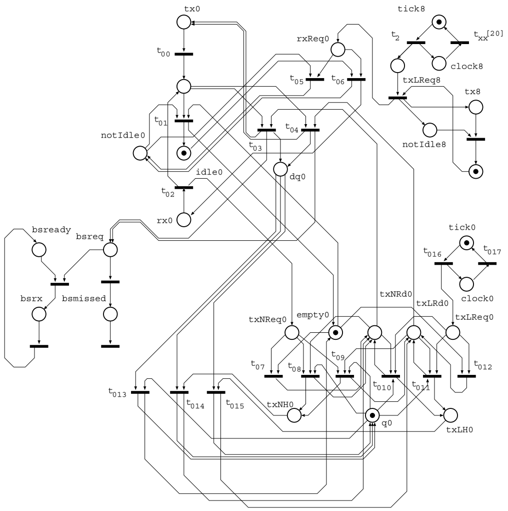


Adaptive Reconfiguration Task

- Purpose: Ensure closed loop SANET performance
- Problem:
  - Control system analysis bounds acceptable end-to-end delay and dropout rates
  - NEST middleware must ensure that sensor data streams to controller/actuator in accordance with these QoS levels
  - Task is complicated by node mobility, congestion, real-world variability
  - This task develops methods to ensure that NEST middleware maintains specified SANET QoS levels
- Approach: SANET initialized, synchronized, with “stable” routing
  - Develop timed Petri net models of SANET routing
  - Reachability analysis of timed Petri net estimates QoS (Lemmon/Buy)
  - Synthesize S/W control structures to enforce QoS levels (Antsaklis)
  - Adaptive re-routing to cope with node mobility (Haenggi)
- SUCCESS METRIC:
  - Model problem’s closed loop performance within specs.
  - Successfully predict NEST middleware QoS (delay/dropout) for model problem
  - Successfully enforce NEST middleware QoS for model problem

Adaptive Reconfiguration – Analysis Task (Lemmon)

- STATUS:
  - Version 2.0 of PARTS toolkit
  - Unfolding tool
  - Analysis of 3-node Petri net model of SANET
  - 2 Journal Papers
- FINDINGS:
  - High-level models of TinyOS components
  - Reachability analysis (PARTS/Unfolding) is feasible: - novel approaches to taming state-space explosion problems are needed



		none	stub	sym	nr+stub
1-2 delay	Node	1208	549	1208	461
	arcs	2210	625	2210	537
	secs	1.2	0.5		0.4
1-5 delay	Node	38804	9415	38804	7840
	arcs	64233	11274	64233	9699
	secs	139.5	13.6		10.5

SUCCESS METRIC

- analysis methods allows us to predict actual QoS achieved by NEST Middleware on Model Problems

Adaptive Reconfiguration – Synthesis Task (Antsaklis)

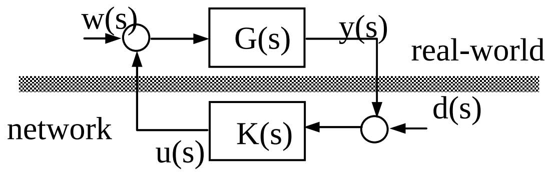
- STATUS:
  - Development of “monitor” supervisor synthesis tool
    monitor is a Petri net controller that can be implemented as a counting semaphore.
  - Trans. Robotics/Automation, Journal paper submitted, 3 conference pubs
- FINDINGS:
  - Existing synthesis tool is well-suited to synthesizing counting semaphores to prevent firing of critical transitions
  - These “semaphore” solutions are not minimally restrictive.
- PLANS:
  - Synthesis tool initially intended for “centralized” problems
    - develop an extension that scales well to distributed problem.
  - Enhanced control structures (beyond semaphores) that provide less restrictive supervision.
- SUCCESS METRIC:
  - Successful generation of supervisors for NEST Model Problem.

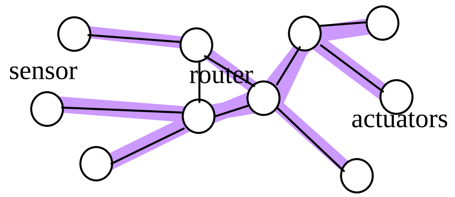
Link Reconfiguration Task (Haenggi)

- STATUS:
  - Development of diffusive routing algorithm
  - NS simulation of adaptive routing
- FINDINGS: too early
- PLAN:
  - NS testing of adaptive rerouting and link reconfiguration algorithms
  - Development of TinyOS components supporting link reconfiguration
  - Implementation of link reconfiguration on model problem
- SUCCESS METRIC:
  - Control duration of time over which connectivity problems result in loss of SANET QoS (delays/dropouts).

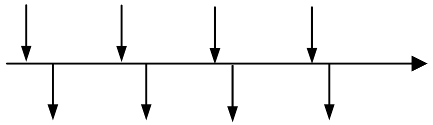
EXAMPLE – QoS Requirements

- SANET is a control system
  Performance means controlling the "energy" of specified output signals in presence of bounded input
- SANET is implemented as distributed computing system
- End-to-end delay's effect on closed loop performance well understood
- Bounds on data dropouts that assure robust performance can be estimated from a stochastic analysis
- We are identifying specific QoS (delay/dropout) levels that assure closed loop performance





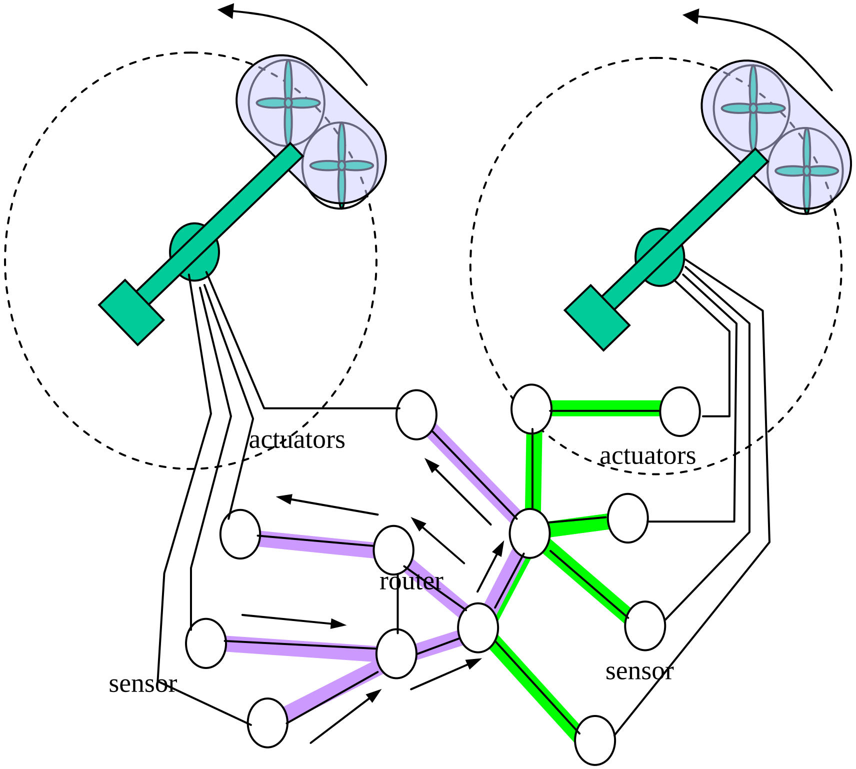
sensor data and transmitted in synchronous manner



data received and processed with "measurable" delays

EXAMPLE – Model Problem

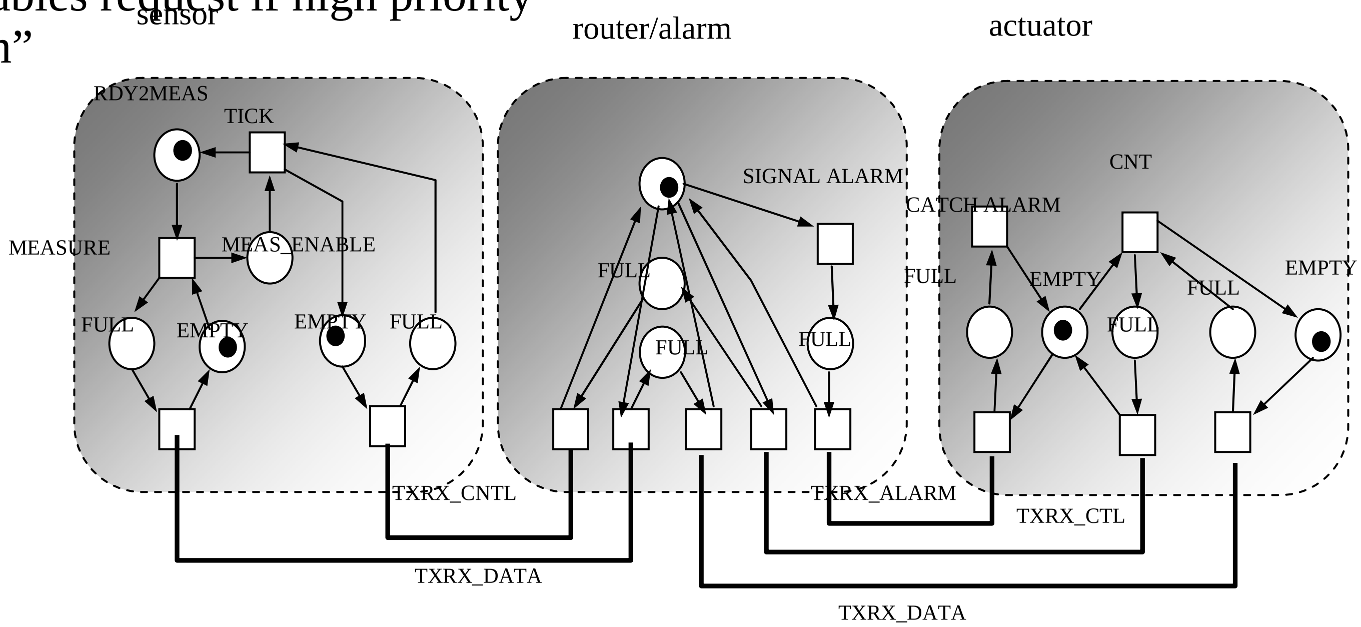
- Remote Control of Multiple Air Vehicles over a NEST built upon Berkeley OEP
- 100 Hz Coordination Frequency
- Current lab setup
  - 2 Quanser 3-dof helicopters
  - 7-node QNX-PC network
- FUTURE PLANS
  - Implement SANET using Berkeley Motes mounted on moveable robotic platforms



EXAMPLE – SANET

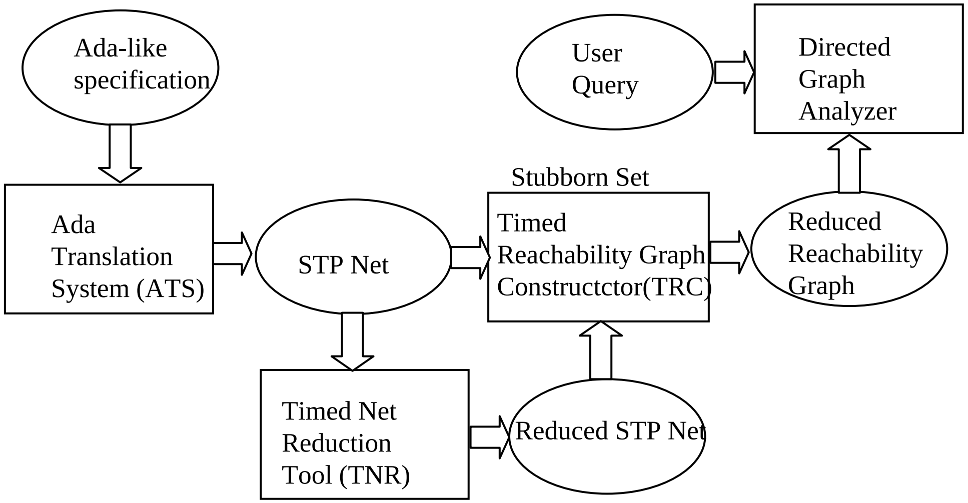
- Sensor Mote
  generates measurement after “request”
- Router/Alarm Mote
  1) forwards data to actuator
  2) generates asynchronous high priority “alarm”
- Actuator Mote
  1) transmits “request” for data
  2) disables request if high priority “alarm”

- Modeling Assumptions
  - TinyOS thread = Timed Transition
  - Frame variables = places
  - MAC protocol abstracted away
  - timed transition is “measurement”
- User Queries:
  - worst case data delivery time
  - worst case response to alarm



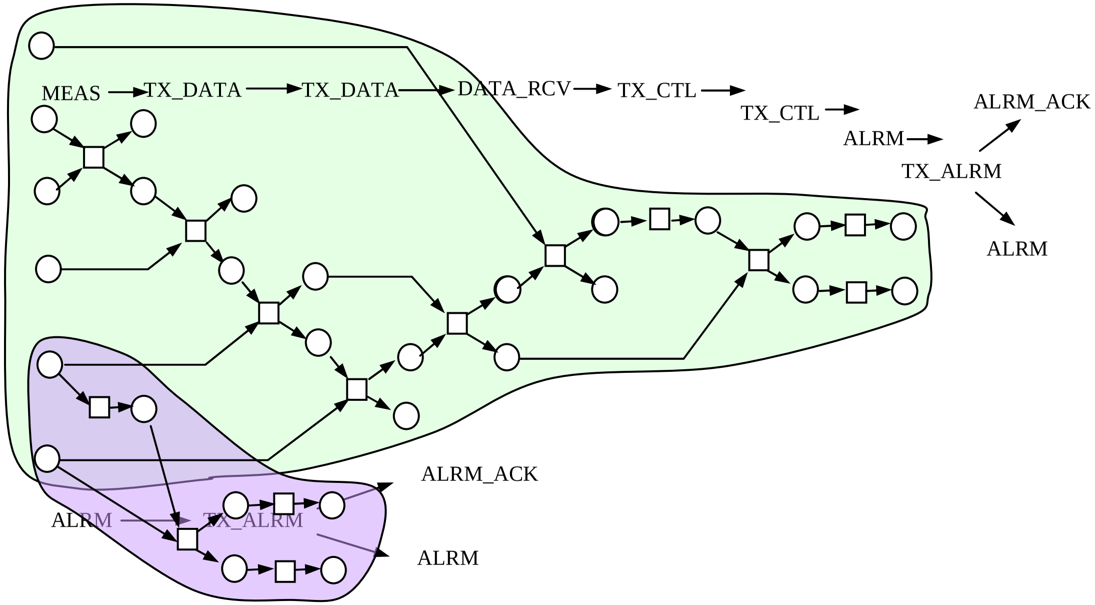
EXAMPLE – SANET ANALYSIS

- Analysis uses state space exploration
  - PARTS (vers. 2.0)
    constructs reachability graph for simple timed Petri net
    uses stubborn sets/symmetry to reduce size of graph
  - PARTS assesses STP system for
    - safety properties (deadlock, mutual exclusion)
    - timing properties (max end-to-end delays)


Ada-like specification
Ada Translation System (ATS)
STP Net
Timed Net Reduction Tool (TNR)
Reduced STP Net
Stubborn Set
Timed Reachability Graph Constructctor(TRC)
Reduced Reachability Graph
User Query
Directed Graph Analyzer

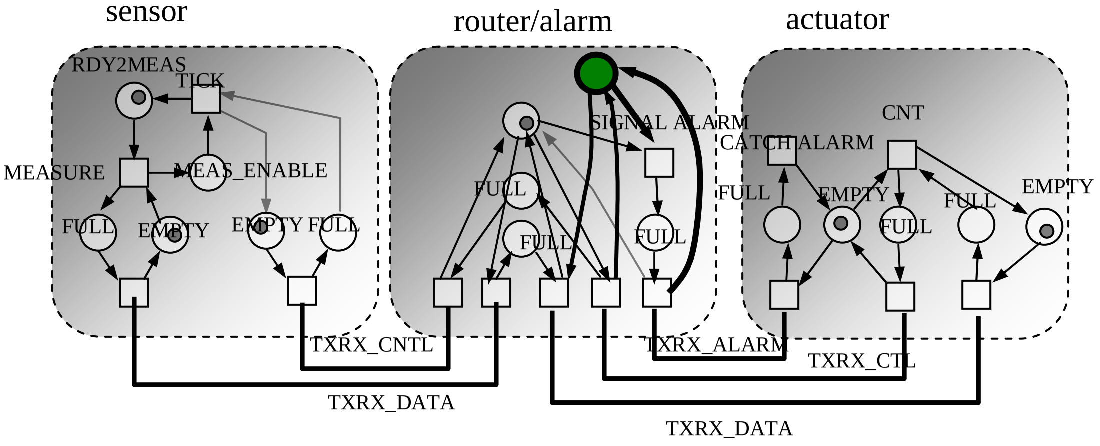
EXAMPLE – SANET ANALYSIS

- Unfolding Analysis Method
  constructs and searches an “event structure” satisfying a branching time semantic.
- Event Structure groups “transitions” into “configurations”
  - conflict-free sets of partially ordered transitions representing a basic execution of the concurrent system.
  - Each configuration is evaluated for safety and timing properties
  - Critical transitions leading to safety violations are identified



EXAMPLE – SUPERVISOR SYNTHESIS

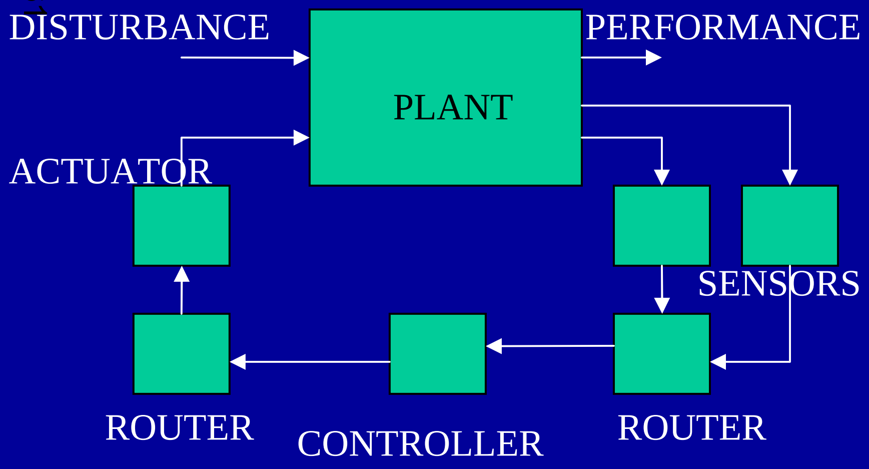
- Critical transitions represent “choices” that the system makes
- Supervisory control involves disabling “bad” choices
- Key technical problem is to synthesize S/W control structures consistent with distributed nature of SANET
  - use of local partial state rather than global full state information
  - respects inherent “controllability” and “observability” of system (e.g. don’t rely on events occurring in two different nodes).
  - implemented using existing S/W control mechanisms
- Example Problem:
  - deadlock occurs when an alarm occurs while actuator’s data buffer full
  - Counting semaphore solution in router/alarm node

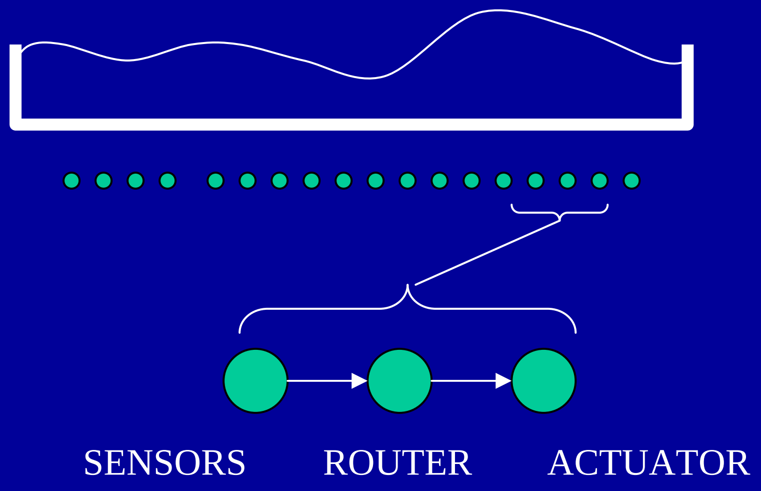


Sensor/Actuator Network

- NEST and Networked Control
- Plan “continuous-time system
- Controller minimizes system sensitivity to disturbances
- Controller is implemented by a network of embedded processors

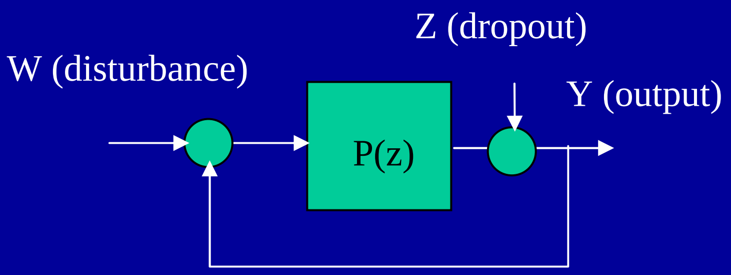
- NEST consists of several closed loop feedback systems
- Distributed vibrational system sensor located at mode peak non-collocated actuator router forwards data packets between sensor and actuator





Sensor/Actuator Network Analysis

- Packet delay and dropout represent primary interface between systems engineer and NEST system designer
- System performance characterized by energy in measured output



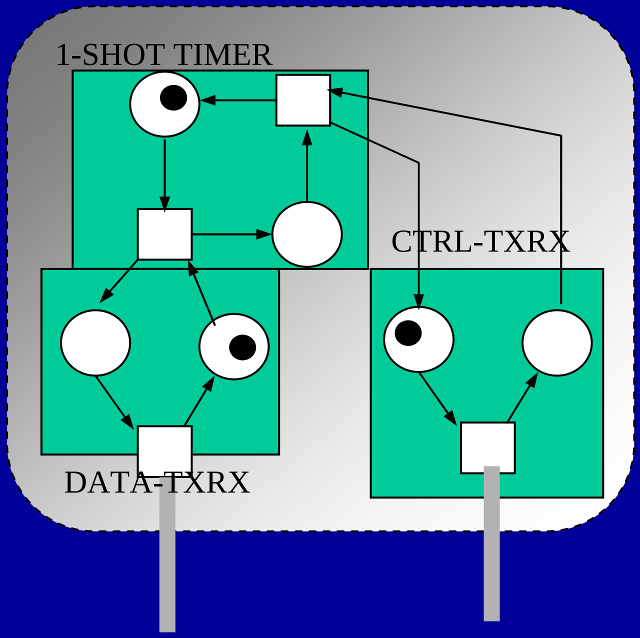
- Packet delay reduces system performance by introducing additional phase lag into system.
- Model effect of packet dropout as a disturbance on overall system performance.
- Performance of perturbed system characterized by equation

$$y = (I + P(z) + \varepsilon D(z))^{-1} P(z) w$$

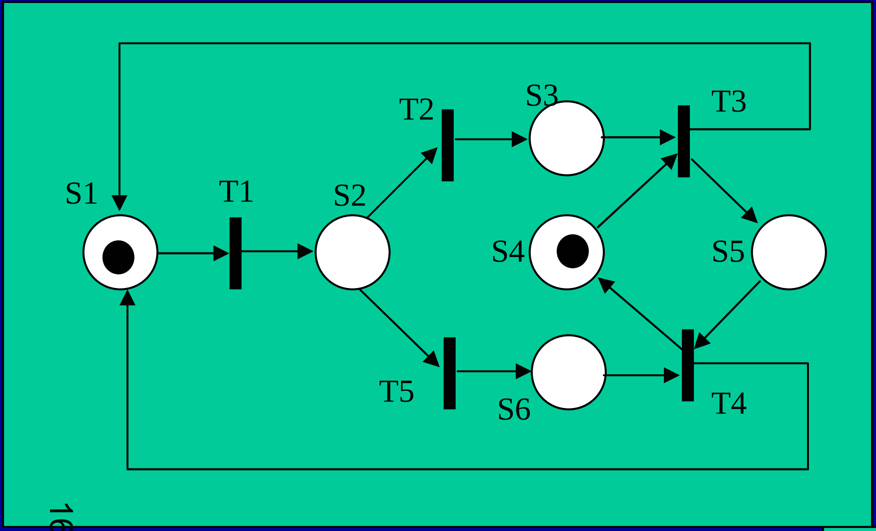
- This allows us to directly relate drop-out probability to overall system performance.
- NEST Middleware must provide guarantees on dropout rate and end-to-end packet delay

Mote Modelling

- Timed Petri Net Model of TinyOS components
- Example shows a simple mote model consisting of 3 components
  - 1-shot timer (sensor measurement)
  - TXRX for data packet
  - TXRX for control packet
- Transitions post threads
- Each transition has a delay time associated with it

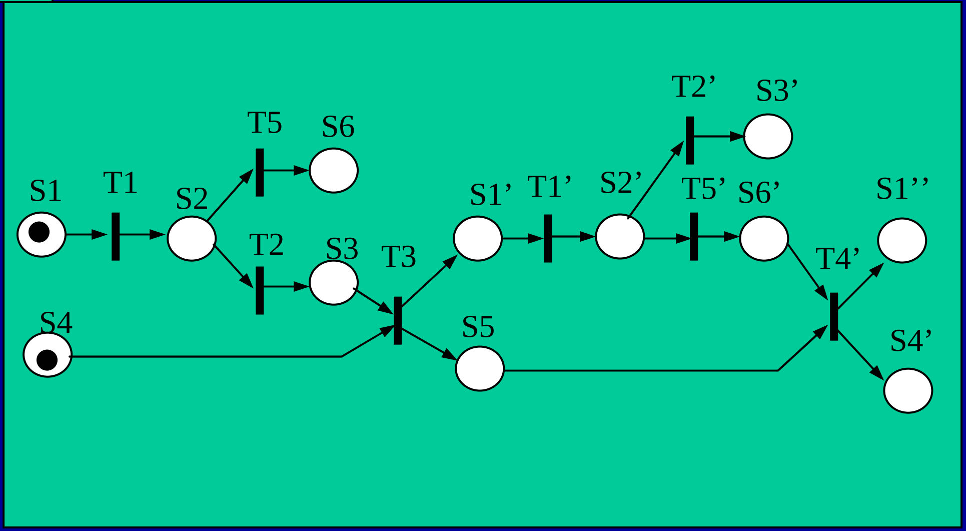


Unfolding Analysis



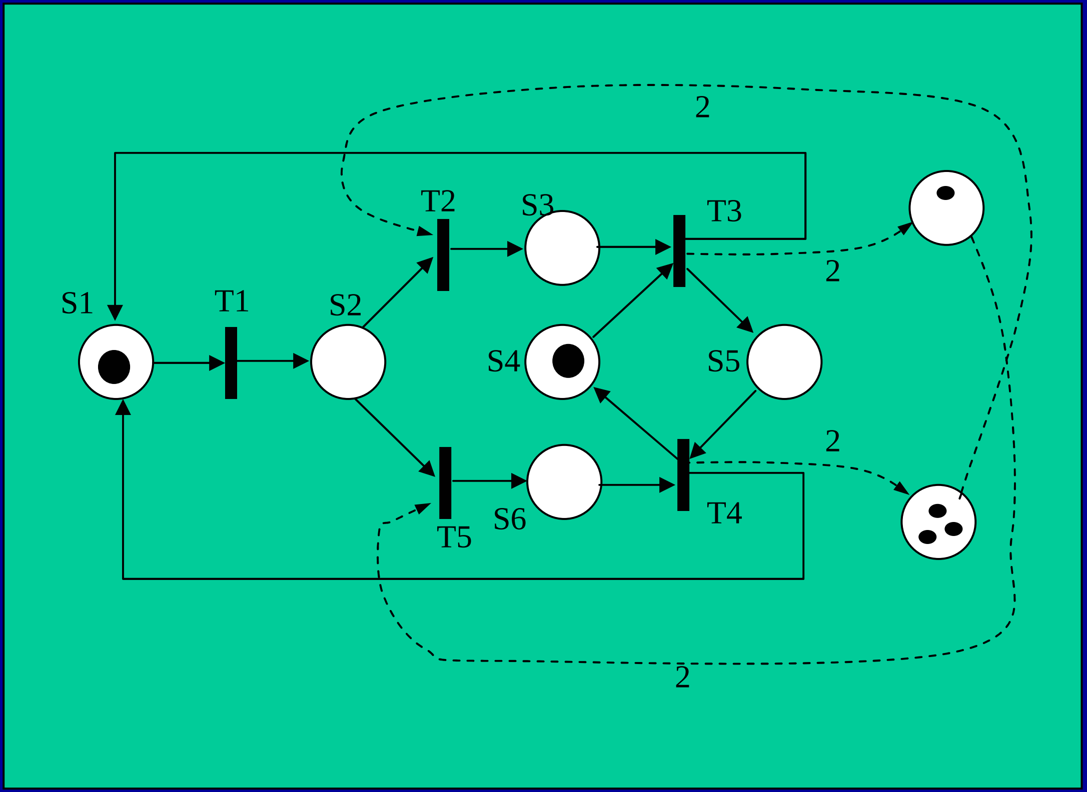
- Formal Analysis transforms original net system into an **event structure**
- Event structure is used to identify **base configurations:** conflict-free sequences of posted threads

- Unfolding analysis can be used to identify deadlocks within net system
- We then construct local supervisor (semaphore to disable deadlock



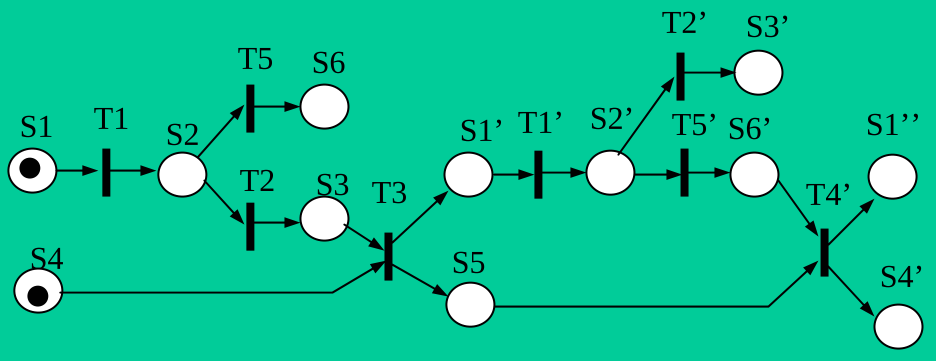
Supervisory Control

- Supervisory Control observes module’s state to adaptively disable unsafe executions.
- Supervisor is a software structure (such as a semaphore) that uses “local” state information to prevent the posting of TinyOS threads.
- Example:
  - Deadlock if choice is executed in wrong order.
  - Two semaphores enforce safe order of execution



Timing Analysis from Unfolding

- Base configurations are formed from conflict free threads.
- All system executions are formed from concurrent or sequential execution of base configurations
- End-to-end delay of a base configuration can be computed using a (max,+) formalism
- Executions with unacceptable delays can also be adaptive disabled by supervisor



THREE base configurations
C1 = { T1,T5}
C2 = { T1,T2,T3,T1’,T2’}
C3 = { T1,T2,T3,T1’,T5’,T4’}

Delay(C1) = T1+T5

Delay(C2) = T1+T2+T3+T1’+T2’

Delay(C3) = T1+T2+T3+T1’+T5’+T4’

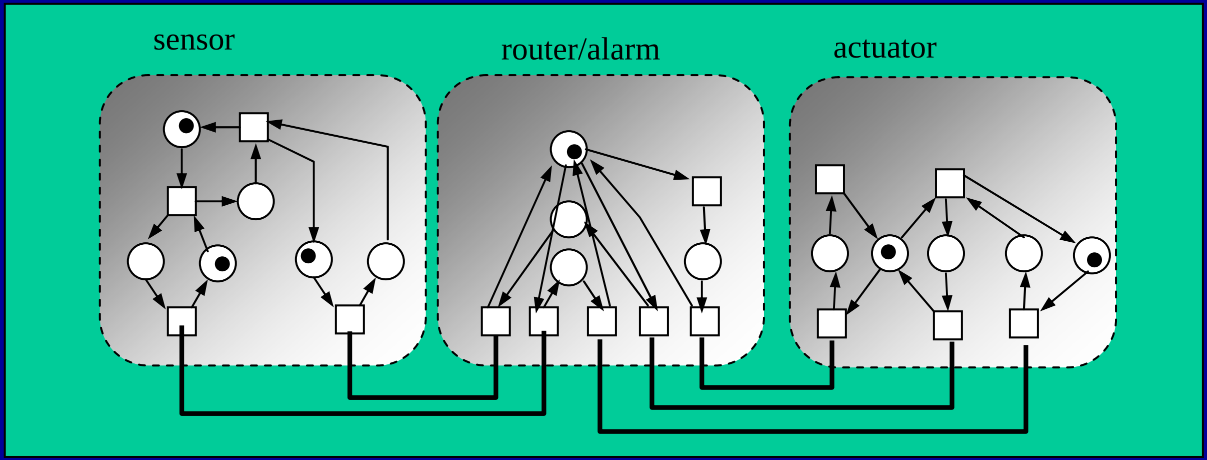
Timing Analysis using PARTS

- Complexity of unfolding analysis is excessive for real-time systems.
- Use of symmetries and stubborn sets can be used to substantially reduce complexity
- PARTS toolit (table)

3 Node Example

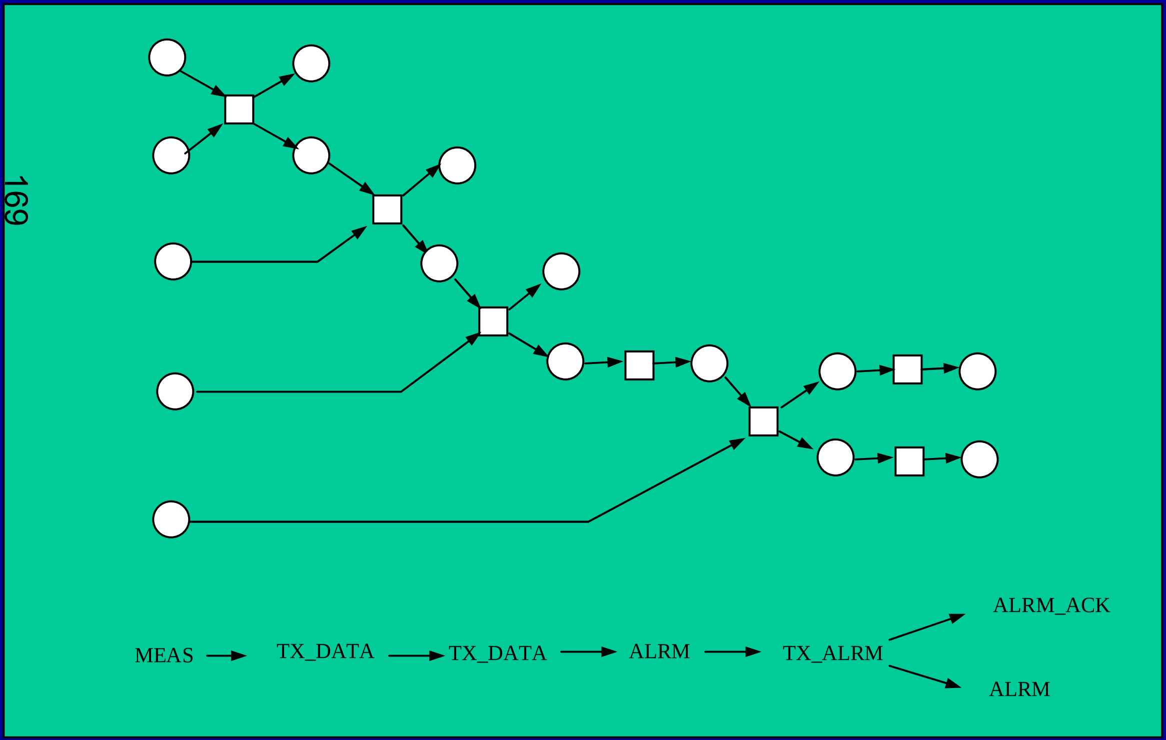
- Sensor Node generates data packet in response to control (ctrl) packet from actuator
- Router/Alarm node must forward data packets from sensor.
- Router/Alarm node also generates alarms that must be caught by the actuator.

- Assumptions:
  - single buffer in router
  - actuator node disables TX of control packet when alarm received
- Objective:
  - safety?
  - delay on data/ctrl packet
  - delay on processing alarm



Original System is Deadlocked

- Partial Order Analysis method was used to extract “base configuration”

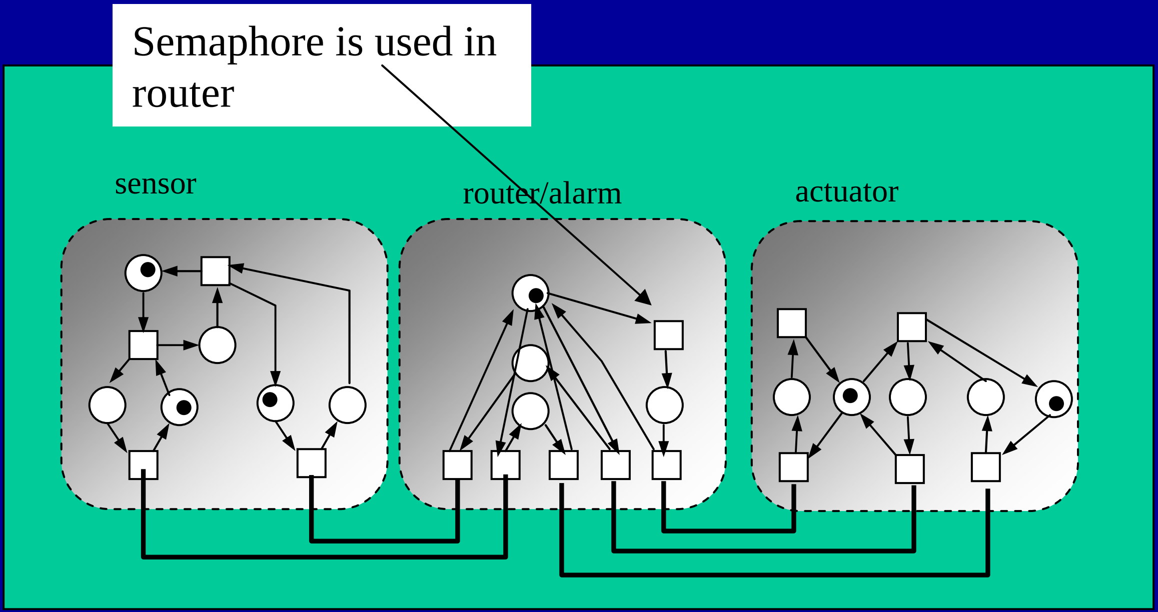


- Unfolding analysis of this system identified 8 base configurations
- It was shown that two of the base configurations were deadlocked.
- Deadlock occurs when alarm is generated and base station receives data packet
- This leads to busy waiting between base station and router

Supervisory Semaphore in Router

- We cannot disable deadlock from the base station, because we don't know when a packet is going to be sent out.
- This is a controllability - observability problem

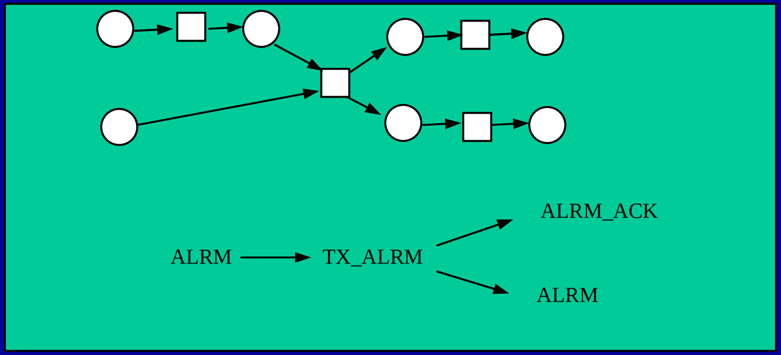
- Use a semaphore in the router to disable the deadlock
- The semaphore
  - disables alarm when a data packet is sent to base station
  - re-enables alarm when control packet is forwarded to sensor

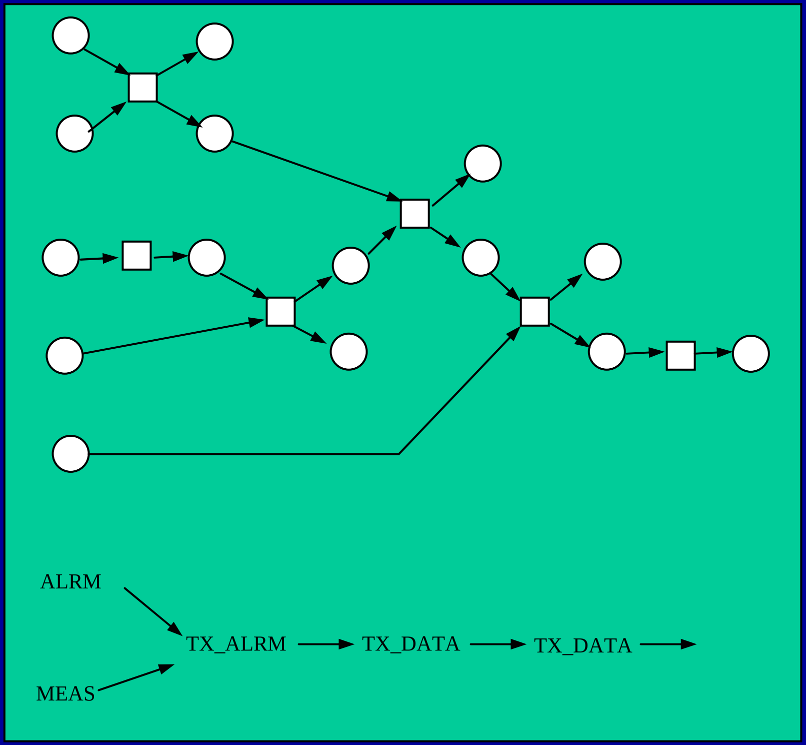


Semaphore is used in router

Delay Analysis

- The supervised system is now live and we can analyze base configurations for end-to-end delay
- Max Alarm Delay =
  time between alarm generation and ack by base station
- Max Data Delay=
  time between generation of data packet at sensor and packet's ack at base station
- Findings:
  - Max Alarm Delay = time to process two alarms
  - Max Data Delay = time to forward data + Max Alarm Delay





Real-time Configuration of Networked Embedded Systems

P.J. Antsaklis (PI)
Dept. of Electrical Engineering
University of Notre Dame

DAPRA/IXO NEST PI MEETING
July 2002, Bar Harbor Maine

Real-time Reconfiguration of Networked Embedded Systems

University of Notre Dame

Contract No. F30602-01-2-0526
AO No: L546/00

Period: 5/2001 – 9/2004
Panos J. Antsaklis (PI)
Department of Electrical Engineering
PI Phone No: 574-631-5792, Fax 574-631-4393, antsaklis.1@nd.edu

DARPA Program Manager: Vijay Raghavan
Agent name & organization: Juan Carbonell, AFRL/Rome

Subcontractors and Collaborators

Senior Personnel

Panos J. Antsaklis (PI), Michael D. Lemmon, Martin Haenggi

Electrical Engineering, University of Notre Dame

Project Consultant

Ugo Buy, Computer Science, University of Illinois-Chicago

Formal models and analysis of embedded system network – Berkeley motes and TinyOS

Graduate Students

Lucia Xia, Luis Montestruque, Xiaowen Liu, Marian Iordache, Hai Lin, Lei Fang, Wenyi Zhang, Brett McMickell

Problem Description and Program Objective

A*daptive reconfiguration in fine grained* networks *of embedded systems* that interact tightly with the physical world so to maintain Quality of Service (QoS.)

Connection “Quality of Service” Metrics

- Delays (max end to end, number of hops)
- Packet dropout rates (link reliability, throughput)
- Power/resource usage (number of messages, CPU cycles)

Contributions to NEST goals

- Projects develops methods assuring connection QoS in the presence of NEST’s ad hoc networking environment.

Success Criteria

- Fast and optimal reconfiguration of ad hoc network connections on model/challenge problems.

Project Task Overview

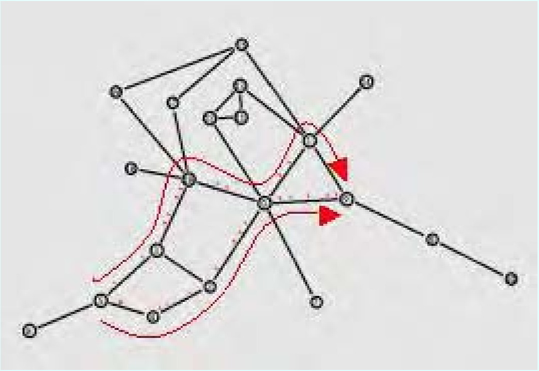
- Clock-synchronization and Security:
  Develop secure clock-sync algorithms for ad hoc networks with drifting clocks.
- Adaptive Reconfiguration
  - Connection QoS Analysis
    - Analysis of timed distributed algorithms.
    - Probabilistic link and network models.
  - Supervisor Synthesis
    - Synthesis S/W control structures to assure connection QoS.
- OEP Integration
  - Delivery of TinyOS components for clock-synchronization, authentication, and dynamic routing.
  - Demonstration of TinyOS NEST services on model problem.

Univ. of Notre Dame

Real-time Reconfiguration of Networked Embedded Systems

Problem and Challenge

Real-time reconfiguration of middleware services in networks of embedded systems



Real-time+ Distributed Algorithms + Feedback Loops+Probabilistic Analysis+Formal Verification

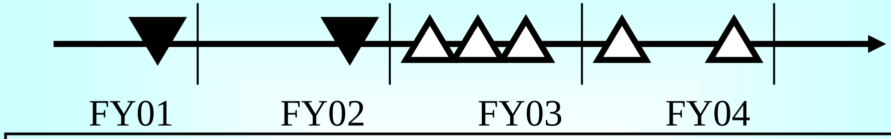
New Ideas

- **Model-based** clock synchronization and security for **heterogeneous networks** of embedded systems
- **Adaptive reconfiguration** of middleware services in response to abrupt network changes
- **Real-time** network management through **probabilistic analysis and formal verification.**
- Exploiting **feedback** in distributed algorithms

Impact

- **Middleware for ad hoc networks** of embedded systems.
- Real-time reconfiguration of **streaming connections** in ad hoc wireless networks.
- Scalable methods for **formation control** in robotic colonies
- Ad hoc networks for **mitigation of seismic** disturbances on buildings.

Schedule



3QFY01: Project Start
3QFY02: Berkeley OEP Demo
1QFY03: Delivery of TInyOS Service Components
2QFY03: Demo of Model Problem (robot formation control)
3QFY03: Integration with Group II components
1QFY04: Challenge Problem Integration
3QFY04: Final Demonstration

Project Status

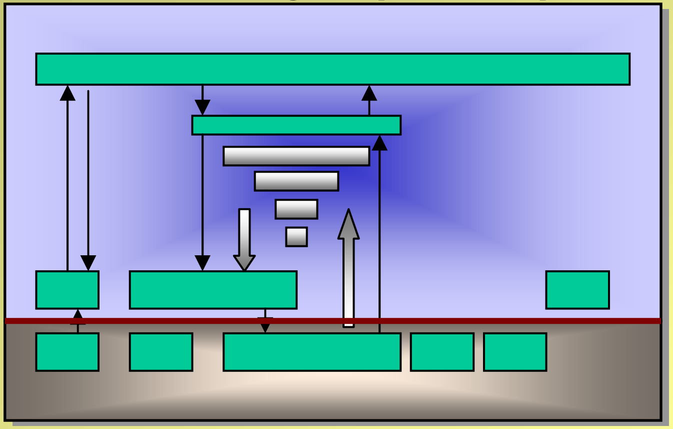
- Mini-task and UCB/OEP demo resulted in addition of probabilistic models to technological approach.
- PROGRESS SINCE LAST MTG.
  - Implementation and Testing of TinyOS components for authentication, clock-sync, route discovery, and dynamic streaming NEST services (UCB OEP DEMO)
  - PARTS (Timed Petri Net Analysis S/W tool) delivered by subcontractor.
  - Design and fabrication of miniature robotic platform using Mica (model problem)
  - Probabilistic analysis of NEST services model
    - probabilistic link model (mini-task)
    - connection quality vs. closed loop performance.

Streaming Service Demo – accomplishments

- Implementation of TinyOS components coordinating execution of multiple NEST services (NEMO)
- Implementation of authentication, clock-synchronization, route-discovery, and streaming services.
- Implementation of telemetry service to monitor network messages and inject commands into the network.
- Development of Java and Matlab tools to visualize network behavior
- Initial testing of AUTH/SYNC and DISC/STRM services on 18-node MICA mote networks.

Demonstration – accomplishments

Standardization of group II components



a set of OS services that coordinated and composed of various services

Clock Synchronization – what was demo'd ?

- Master vs local clocks ? – LOCAL
- Sync of heterogeneous network.
- TOSSIM predictions
  - 7 msec sync errors
  - formation of sync clusters
- Experimental testing of demo
  - 20 msec sync errors (30-50 percent)
  - formation of clusters
- Cross-layer implementation will reduce sync errors to micro-second range.

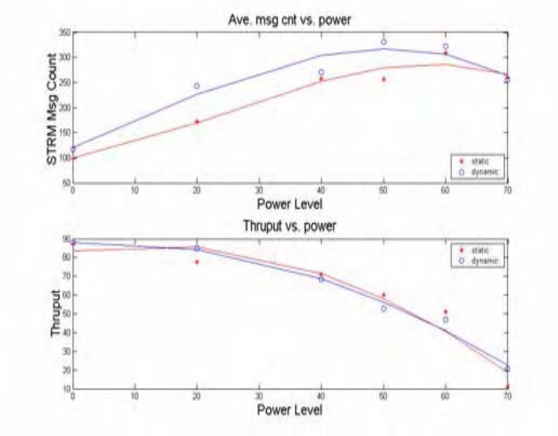
Route Discovery – spanning trees

- Single hop vs Multi-hop? – multi-hop
- Node's a priori knowledge of the network topology? - NONE
- How does one measure scaling?
  - Number of nodes in a cell
  - Number of cells in a network
  - Number of hops on a connection
  - Number of open connections

Streaming service

Experimental results that provided some new conjectures abou the cost of reconfiguration.

Verification of fast connection re-routing

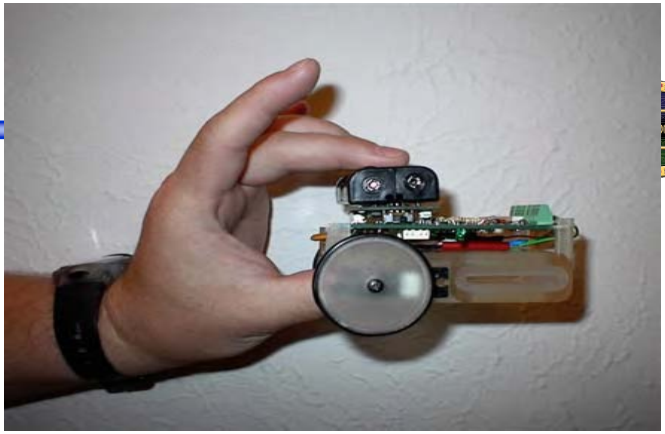


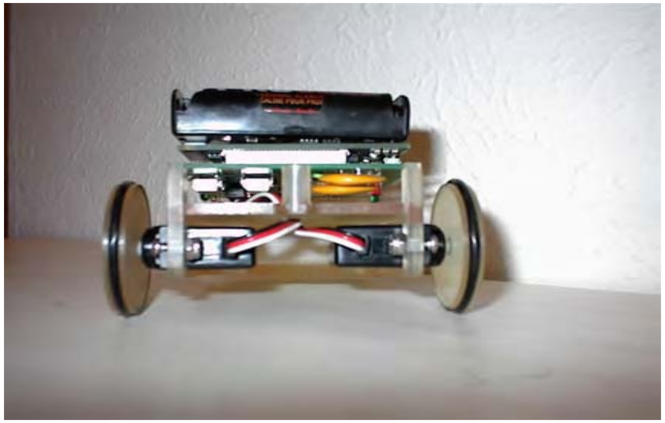
Streaming Service Demo – future plans

- Principal Findings
  - Probabilistic nature of links has important effect on performance of distributed algorithms.
  - Maximizing service performance will require cross-layer protocols
- Future Plans
  - Complete integration of all ND’s NEST services and test on large 36-72 node network. Delivery of NEST middleware and NEMO components along with documentation.
  - Integration of NEST service components with a model problem (robotic formation control).
  - Analytical validation of Empirical Results
    - PARTS toolkit analysis.
    - Probabilistic analyses.

Miniature Robotic Platform accomplishments

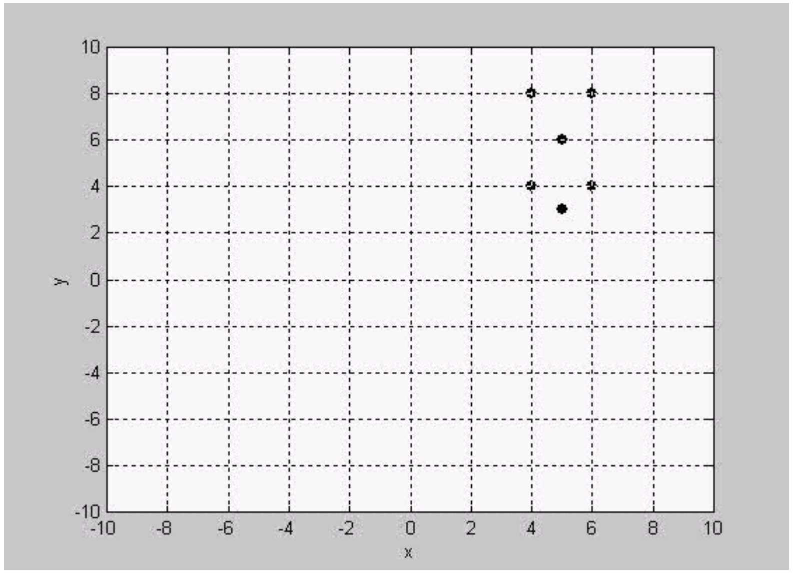
- Mobile robots to investigate group formation and pursuit-evasion game algorithms
- Robot design based on the Berkeley MICA mote.
- Low-cost robotic platform – $210 per robot.
- Small size - 3.25” by 2.75”.
- Hall-Effect sensors for dead reckoning odometry.
- ND designed I/F board.
  - Two optically isolated H-bridges drive the motors using independent PWM channels from MICA.
  - Four optically isolated digital inputs connected to hall-effect sensors
  - Independent power supplies for Mote and power electronics
  - Two ADC inputs for other types of sensors
- Modified UCB Radio Stack frees up PWM and input capture interrupts.





The Model Problem (future plans)
Large-Scale Robotic Formation Control

- Goal: Design and experimentally validate fully scalable formation control algorithms for groups of distributed robots.
- Formation control algorithms are based on a symmetric reduction of the state-space.
- The reduction technique decreases the computation and memory requirements for formation control.
- Reduction is necessary for generating formations on memory constrained platforms such as the Mica mote.
- NEST Middleware Requirements
  - **Time Synchronization Service**
  - **Neighborhood Maintenance Services**
  - **Data Exchange Service**
  - **Local Coordinate System Service**



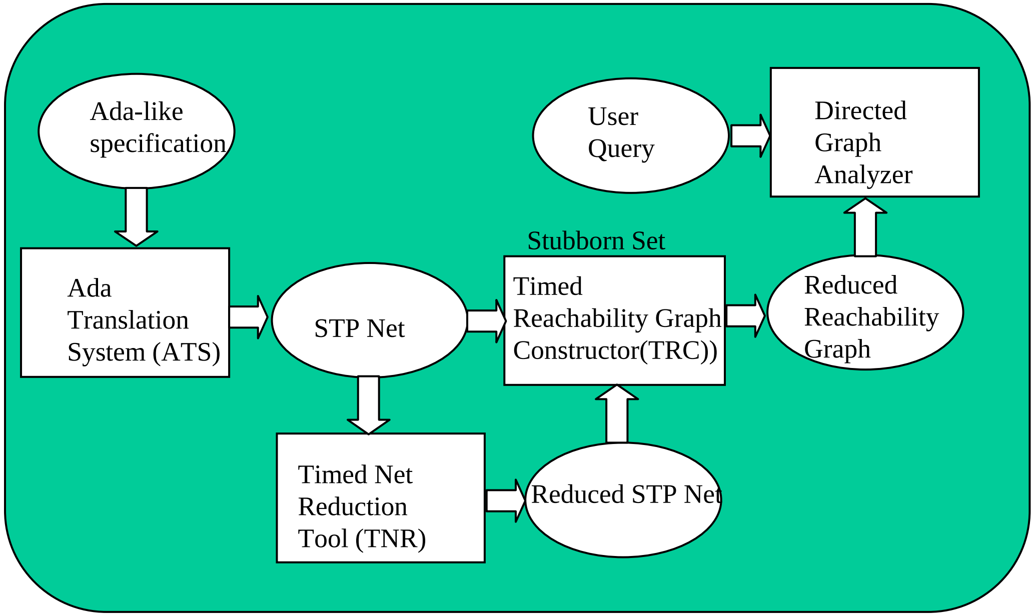
In this simulation, 6 robots form a circle at the center of the graph. Each robot uses only local information about neighboring robots throughout the simulation. The method used in this simulation is based on [1].

[1] H. Yamaguch, “Adaptive formation control for distributed autonomous mobile robot groups”, IEEE ICRA, 1997

PARTS – accomplishments and future plans

- Subcontractor (Ugo Buy) delivered PARTS toolkit (May 2002)
  - C-LISP tool that analyzes simple Timed Petri nets using a variety of reduction and stubborn set methods.
  - assesses simple timed Petri net system for
    - safety properties (deadlock, mutual exclusion)
    - timing properties (max end-to-end delays)

- Future Plans
  - Petri net models of Notre Dame's NEST services
  - PARTS analysis of services
    - sync service instability

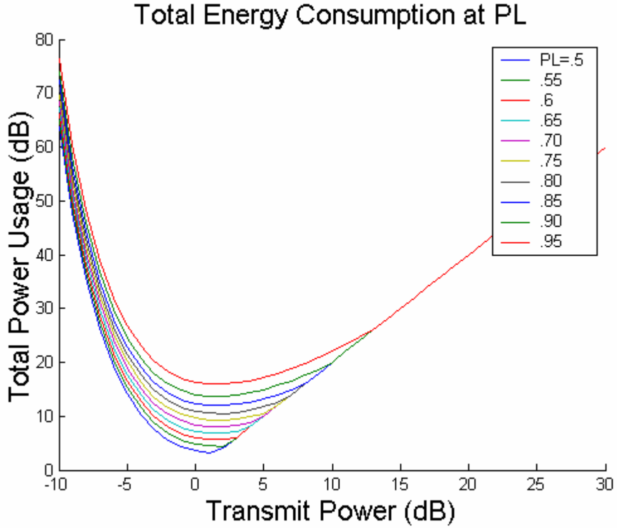

Ada-like specification
Ada Translation System (ATS)
STP Net
Timed Net Reduction Tool (TNR)
Reduced STP Net
Stubborn Set
Timed Reachability Graph Constructor(TRC))
User Query
Directed Graph Analyzer
Reduced Reachability Graph

Probabilistic Link Analysis accomplishments

- Accomplishments
  - Mini-task developed probabilistic link model [1] (Rayleigh fading channel)

$$\text{reception probability} = p_r = \underbrace{e^{-\Theta/\gamma_n}}_{\text{noise}}\underbrace{e^{-\Theta/\gamma_i}}_{\text{interference}}$$

- ▪ Independence of noise and interference analysis
- ▪ Total power consumption is minimized for TX power that is only 1.6 dB above noise level
- ▪ Optimal Power usage is independent of desired link probability, P_L

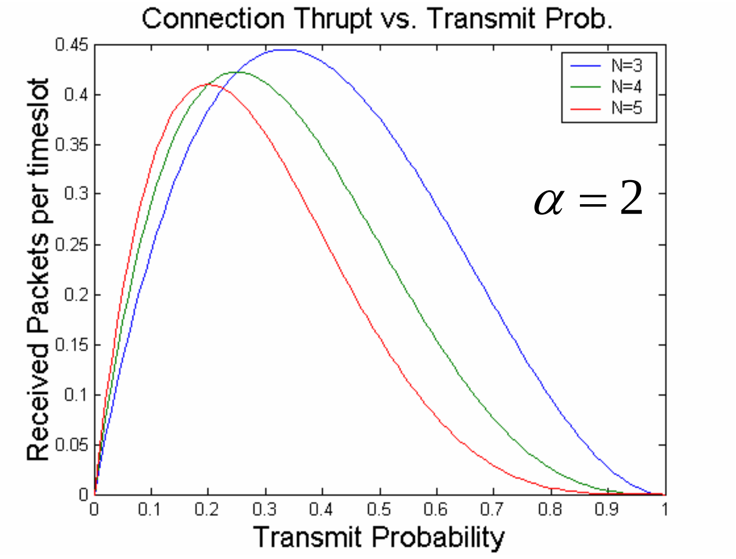


[1] M. Haenggi, Performance Analysis of Ad Hoc Wireless Networks based on a Probabilistic Link Model, INFOCOM 2003 (submitted)

Connection Analysis accomplishments/future plans

- Link model can be used to analyze connection probability
  (N=route length, p = transmit probability)

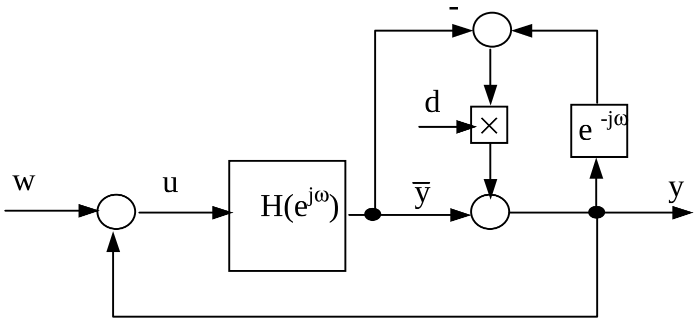
$$g_N(p) = \sum_{k=1}^{N-1} c_k p^k (1-p)^{N-k}$$

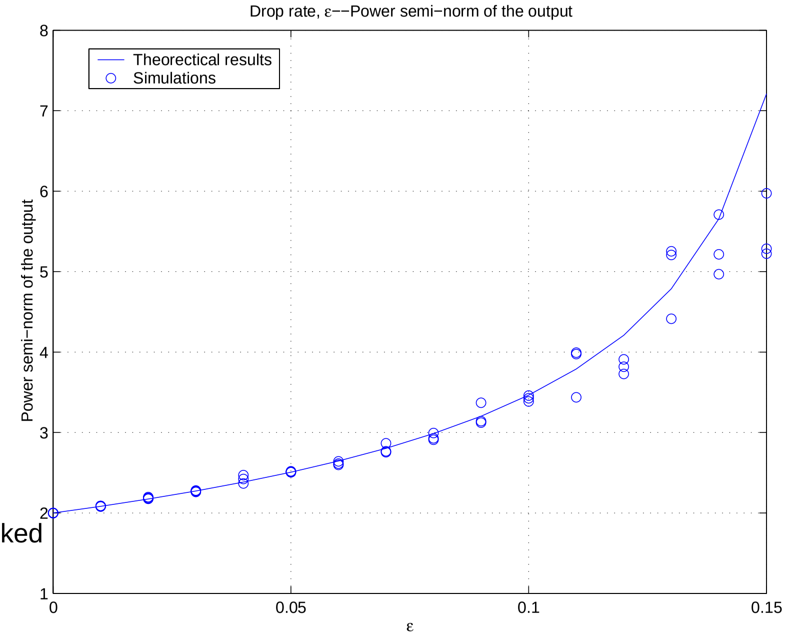


- Future Plans
  - Use link model to analyze DISC/STRM service
  - Validate link model against physical system.

Control Loop Performance accomplishments/future plans

- Accomplishments
  - Closed form expressions for closed loop performance as a function of connection reliability (dropout rate) [1]
- Future Plans
  - Apply to automatic TX power control and congestion control.
  - Synthesize specifications for optimal soft scheduler.





[1] Q. Ling and M.D. Lemmon, Robust Performance of Soft Real-time Networked Control Systems with Data Dropouts, IEEE CDC, 2002 (submitted)

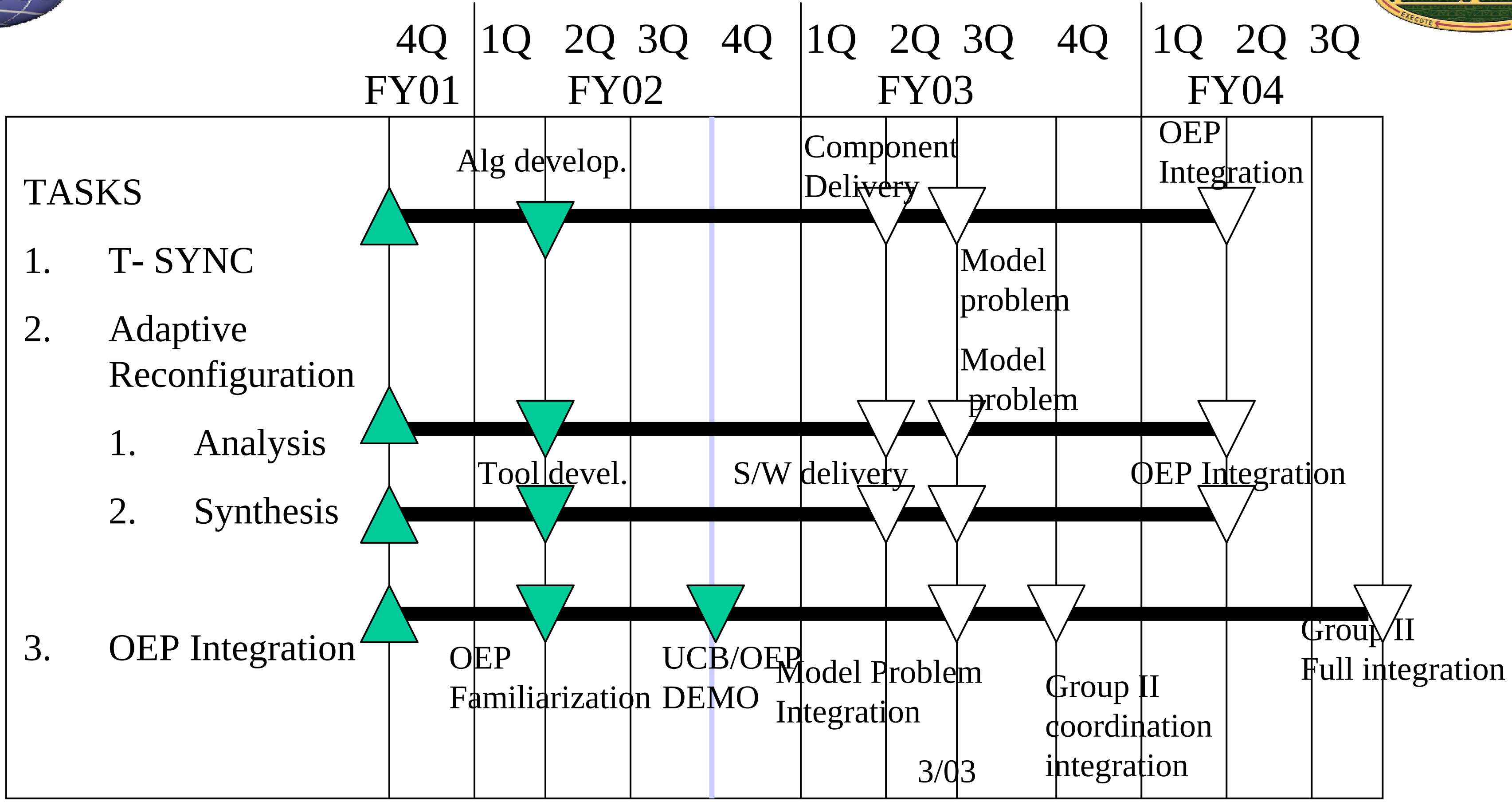
OEP Goals and Success Criteria

- Working with Berkeley OEP
  - Deliver NEST middleware service components
  - Provide real-time O/S components coordinating NEST services with application services
  - UCB/OEP Point of Contact = Dave Culler
- Goals and Success Criteria
  - High Security and Precise Synchronization Services
    - Small sync errors (<10 msec) across at large portion of the network.
    - High probability of detecting malicious users
  - Fast and Optimal Reconfiguration of Connection Service
    - High throughput (80 percent) with bounded message overhead.
  - Successful Model Problem Demonstration
    - Coordinated movement of 5-10 robots
    - Low average tracking error and high coordination frequency

Project Plans and Goals

- Plan over next 6 months
  - Delivery of sync/auth/disc/strm service components.
    - CVS tree with documentation
    - 10 msec sync errors and 80 percent throughput for streaming demo on network of 36-72 nodes.
  - Prototyping of Model Problem (Multi-robot Formation Control).
    - TBD tracking error vs TBD coordination frequency
  - Probabilistic analysis of connection services.
    - prediction and validation of actual OEP connection service throughput
    - development of congestion control or TX power control service.
  - PARTS analysis of connection services.
    - validation of sync algorithm’s instability
    - automated synthesis of “supervisory controller”

Schedule



- On schedule to meet 12/02 milestones.
- Milestone: Demo of NEST services on UCB’s OEP.

Technology Transition/Transfer

- Providing middleware services to a variety of research groups at Notre Dame that are using Berkeley OEP
  - Seismic mitigation, formation control, real-time systems.
- Developing additional I/F and spread-spectrum communication boards that can extend abilities of the MICA platform to support projects in these various groups.

Program Issues

- Standards for TinyOS component delivery?
- 10 percent failure rate on initial shipment of MICAs!
  - additional 10 percent failure over last 6 months.
- High cost of additional programming boards and Rene sensor boards!

Real-time Reconfiguration of Networks of Embedded Systems

Data Streaming Service in Ad Hoc Wireless Networks

DARPA/IXO NEST MEETING
Berkeley OEP Demonstration
July 2002, Bar Harbor Maine

P.J. Antsaklis (PI)
University of Notre Dame

Real-time Reconfiguration of Networked Embedded Systems

University of Notre Dame

Contract No. F30602-01-2-0526
AO No: L546/00

Period: 5/2001 – 9/2004

Panos J. Antsaklis (PI)
Department of Electrical Engineering
PI Phone No: 574-631-5792, Fax 574-631-4393, antsaklis.1@nd.edu

DARPA Program Manager: Vijay Raghavan
Agent name & organization: Juan Carbonell, AFRL/Rome

Subcontractors and Collaborators

Senior Personnel

Panos J. Antsaklis (PI), Michael D. Lemmon, Martin Haenggi

Electrical Engineering, University of Notre Dame

Project Consultant

Ugo Buy, Computer Science, University of Illinois-Chicago

Formal models and analysis of real-time systems

Graduate Students

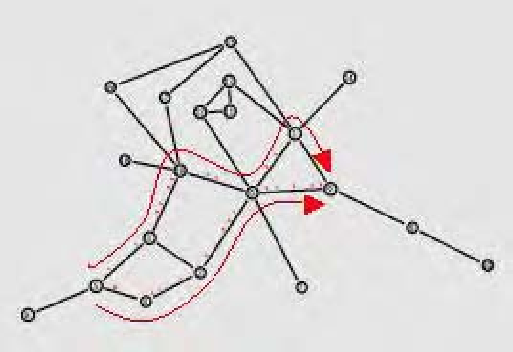
Lucia Xia, Luis Montestruque, Xiaowen Liu, Marian Iordache, Hai Lin, Lei Fang, Wenyi Zhang, Brett McMickell

Univ. of Notre Dame

Real-time Reconfiguration of Networks of Embedded Systems
Data Streaming in Ad Hoc Wireless Networks

Demonstration summary

Data Streaming Service on the Berkeley OEP



Real-time reconfiguration of data connection in response to abrupt changes in network connectivity

Relationship to project

- Coordination of low-level NEST services: authentication, synchronization, route-discovery, streaming, telemetry
- Authentication used to form secure neighborhoods
- Synchronization used to coordinate collective actions
- Route-discovery used to construct routing tables
- Streaming used to transmit large amounts of data in a fixed time interval
- Telemetry service used to monitor NEST behavior

Relationship to Berkeley Challenge Problem

- Demonstrated services of use to the challenge problem
  - ✓ authentication
  - ✓ clock-synchronization
  - ✓ route discovery
  - ✓ data streaming
  - ✓ telemetry
- Complementary services needed for challenge problem
  - ✓ localization
  - ✓ neighborhood maintenance

Evaluation Criteria

- Demonstration Metrics:
  - ✓ Synchronization of network with drifting clocks.
  - ✓ Fast re-routing in presence of node failures
- NEST service Metrics
  - ✓ Synchronization Errors (< 50 msec)
  - ✓ Connection throughput (> 80 percent)
  - ✓ Message Usage

Overview

Data Streaming in Ad Hoc Wireless Networks

- Introduction
  - Project Synopsis
  - S/W architecture used to coordinate NEST services
- Demonstration
  - Authentication/Synchronization Demo
  - Discovery/Streaming Demo
- Relation to Challenge Problem
- Evaluation Criteria
- Experimental Evaluation of Service Performance
- Demonstration Issues
- Lessons Learned

Project Synopsis

Real-time Reconfiguration of Networks of Embedded Systems

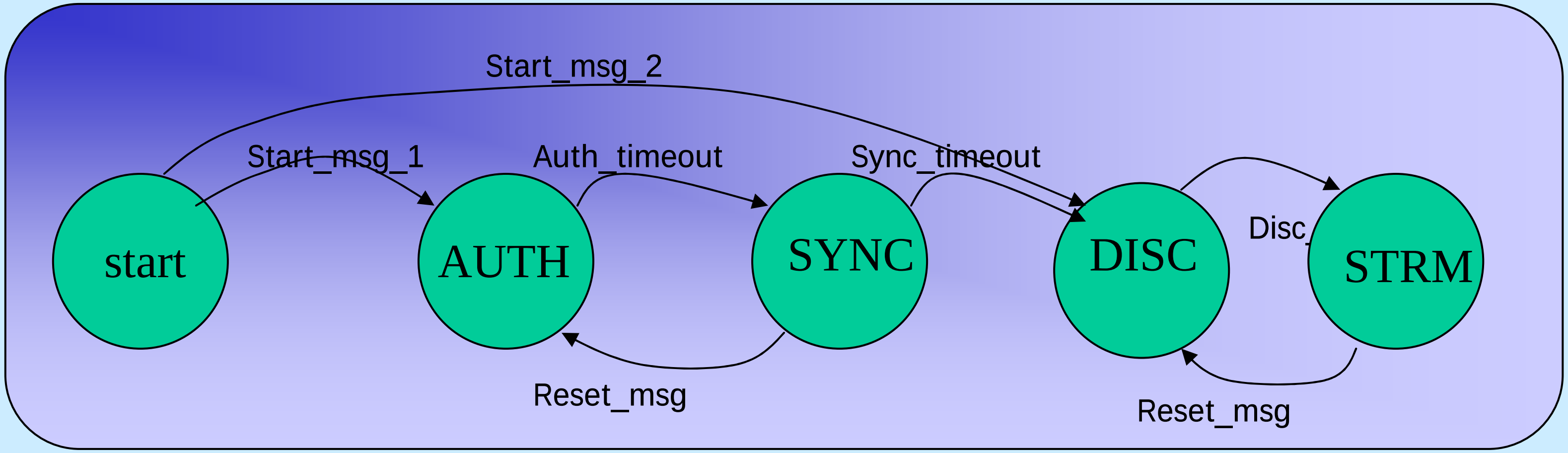
- A*daptive reconfiguration in fine grained* networks *of embedded systems* that interact tightly with the physical world so to maintain Quality of Service (QoS.)
- Project Tasks:
  - Clock-synchronization and Security:
    develop secure clock-synchronization algorithms for ad hoc networks with drifting clocks.
  - Adaptive Reconfiguration
    Fast reconfiguration of NEST services in the presence of abrupt network changes.
  - OEP Integration
    Currently working with Berkeley OEP.

Data Streaming Service

- Streaming is a high-level NEST service that reliably transmits a large block of data in a fixed period of time over an ad hoc wireless network.
- Streaming is built up from low-level NEST services
  - **Authentication** identifies a secure/reliable neighborhood for each node.
  - **Clock Synchronization** enables synchronization of neighborhood clocks to facilitate coordinate actions between network nodes.
  - **Route Discovery** provides each node with a “next-hop” routing table that points to a terminus node.
  - **Static/Dynamic Streaming** opens and maintains the streaming connection in the presence of network changes.
  - **Telemetry** provides a mechanism for monitoring and supervising the entire network.

Node’s Life Cycle

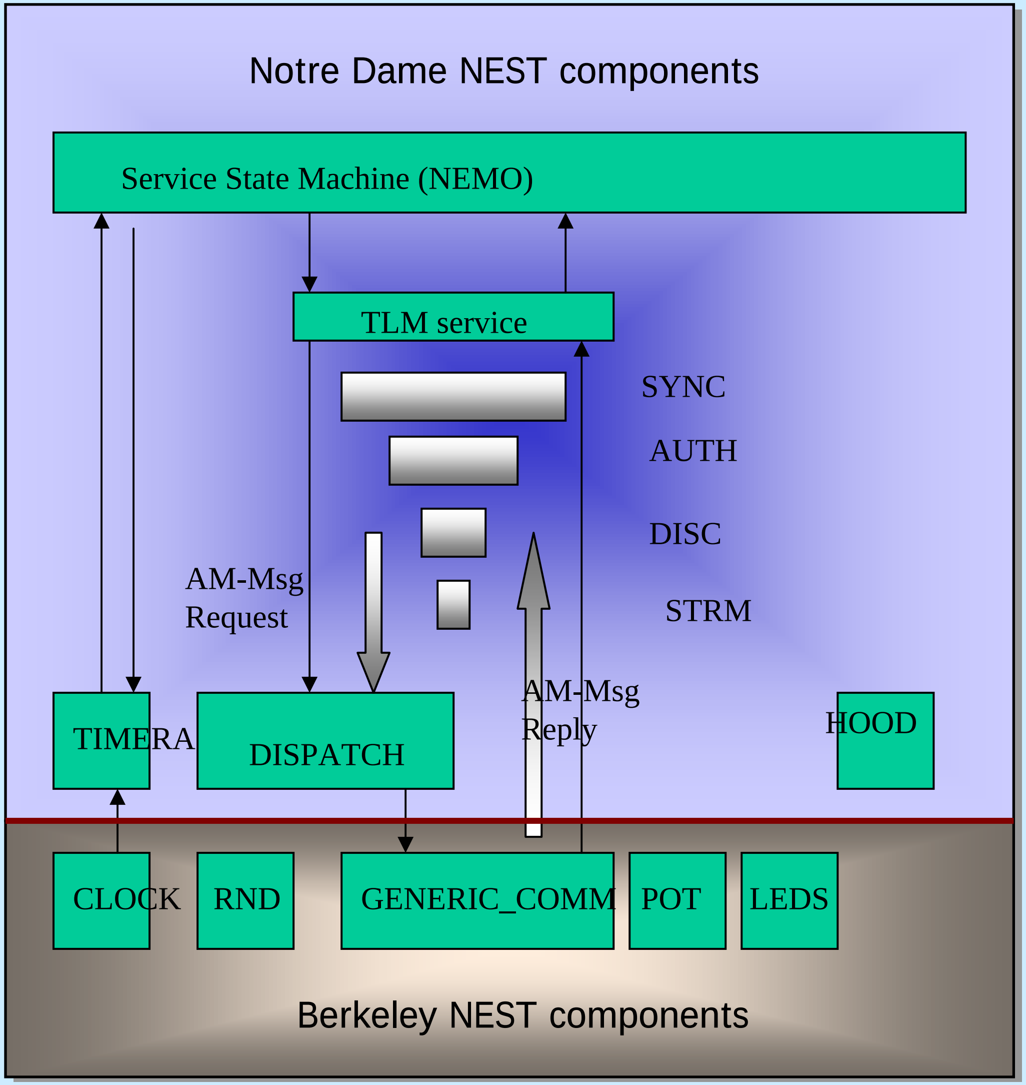
- Execution of low-level NEST services must be coordinated to systematically acquire information required by a connection



- Node progresses through a sequence of “states”
- Each “state” corresponds to dominant/active service.
  - Active services periodically transmit data/request
  - Passive services simply listen for messages.
- At a given time instant, network nodes will not be in the same state. This means all services must be enabled.

Notre Dame's NEST S/W Architecture

- OS components used to realize streaming service
- TinyOS Scheduling Comp.
  - NEMO (service state machine)
  - TIMERA (service scheduling)
  - DISPATCH (msg dispatch)
- TinyOS data object
  - HOOD (neighborhood data)
- NEST Services
  - TLM (telemetry)
  - SYNC (clock-synchronization)
  - AUTH (authentication)
  - DISC (route-discovery)
  - STRM (data-streaming)
- ND's comps. built on top of Berkeley's H/W comp.



Notre Dame's NEST Demonstrations

Authentication/Synchronization Demo

- Authentication Service estimates clock drift/offset to form secure neighborhoods
- Synchronization Service forces nodes in the network to "act" in synchrony
- Demonstration Script:
  - ✓ Start Network in AUTH state
  - ✓ Message traffic discussion
  - ✓ View sync'd network
  - ✓Use VisTool to visualize links
  - ✓Identify clusters of sync'd nodes
- Demonstration Metric: Majority of Network Nodes Blink their LEDS in Synchrony.

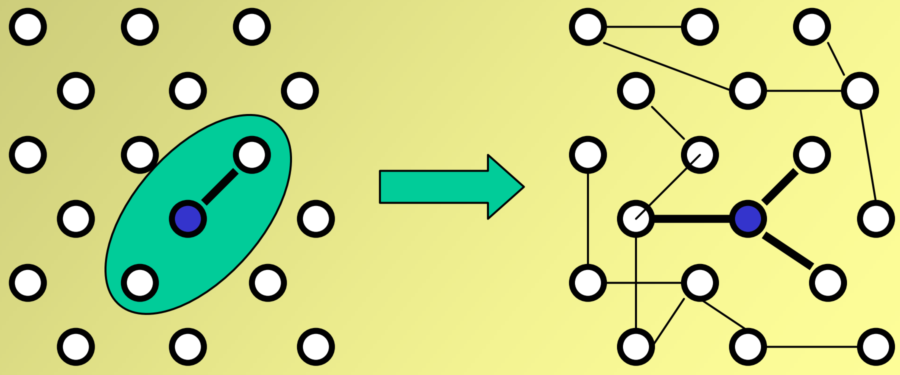
Discovery/Streaming Demo

- Discovery Service constructs min-hop routing table to specified terminus
- Streaming service transmits 100 packets at 1 packet per second.
- Demonstration Script:
  - ✓ All nodes start with all LEDs on
  - ✓ LEDs change to show min. hops to terminus
  - ✓ Start streaming and identify on-route nodes (counting LEDs)
  - ✓ Reduce power of on-route node and observe re-routing of connection
- Demonstration Metric: successful re-routing of connection

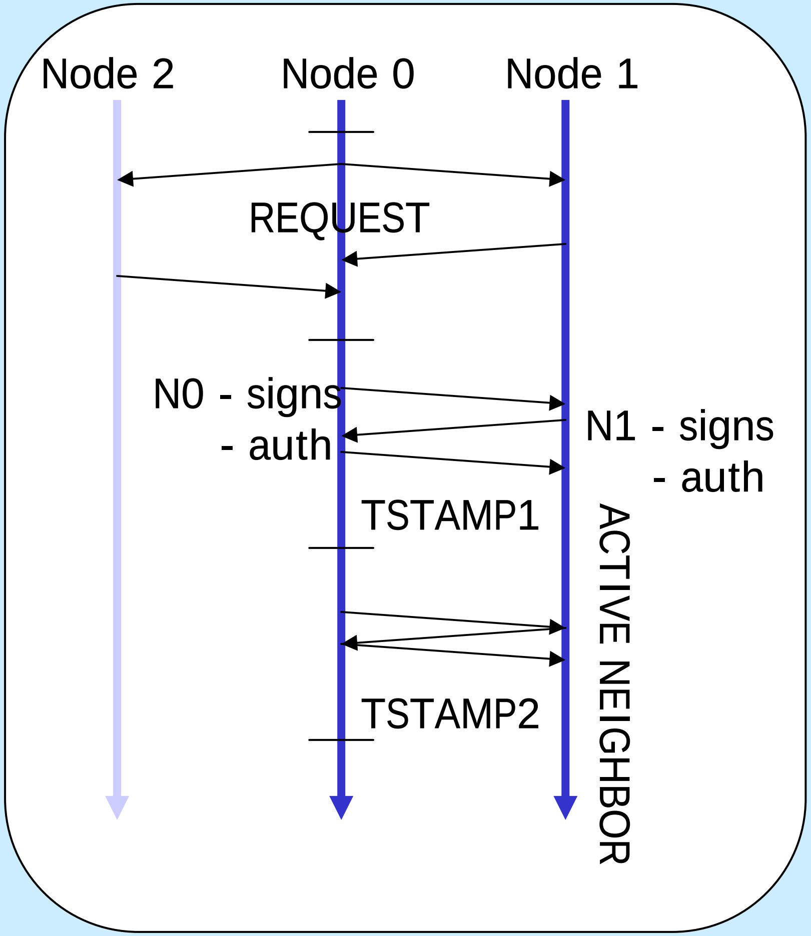
Authentication Service

Authentication Service

- Node requests a “link” to a neighbor
- Exchange of time-stamped messages
  - Secure 3-way handshake (not imp.)
  - estimates of neighbor drift/offset after two exchanges
- Successful link is stored in HOOD



Authentication as a random graph process



Packet Fields – AUTH/SYNC Demo

AUTH process:

7e 0 b 7d 5a 43 0 0 78 5f 0 0 77 52 0 0...... at *15152*

type sender ID receiver ID Localtime 1 *sending time at PC (msec)*

7e 0 c 7d 78 5f 0 0 5a 43 0 0 3a 55 0 0 77 52 0 0 ......

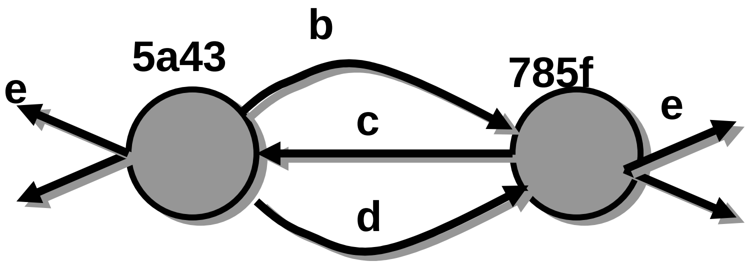
Remote Tstamp

7e 0 d 7d 5a 43 0 0 78 5f 0 0 7d 52 0 0 3a 55 0 0 77 52 0 0 40 fd ff ff......

Tstamp_2 clock offset

7e 0 e 7d 78 5f 0 0 5a 43 0 0 c0 2 0 0 **f4 1 0 0** **f4 1 0 0** c0 2 0 0 ......

numerator/denominator : clock drift



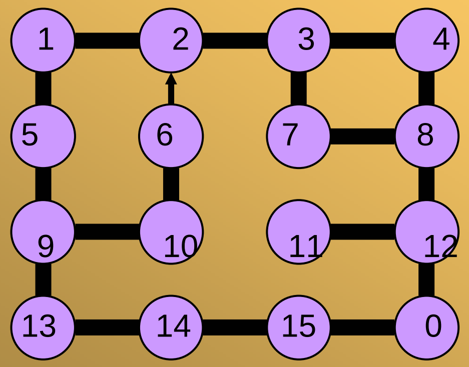
Clock Synchronization Service

- Each node runs two tasks.
  - TIME_MONITOR task periodically broadcasts SYNC_MSG with SYNC_COUNT and SYNC_PERIOD
  - MSG_HANDLER task listens for SYNC_MSG and checks to see if
    1) local SYNC_COUNT is late
    2) remote SYNC_PERIOD is shorter
    and then resets its variables and broadcasts its own SYNC_MSG
- Self-stabilizing Clock Synchronization Alg. (Dolev/Halpern/et al.) for networks with no clock drift.
- By transforming received SYNC_PERIOD using estimated clock drift/offset, the Dolev/Halpern algorithm can be extended to networks with drifting clocks.

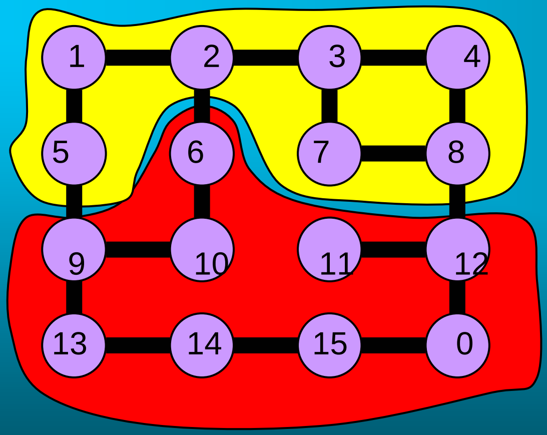
Synchronization Clusters

- Synchronization seeks to get all nodes doing the same thing at the same time.
- Experiments demonstrate a clear clustering phenomenon
- Clustering seen in TOSSIM as well as H/W experiments.
- Hypothesis is that sync clusters represent a “stable” state of the network.

Authentication result (TOSSIM)



Clock Synchronization Result (TOSSIM)



NODE	LED_TIME
5	5: 22.1126
1	5: 22.1135
4	5: 22.1157
7	5: 22.1168
3	5: 22.1178
2	5: 22.1206
8	5: 22.1228

NODE	LED_TIME
12	5: 22.1608
11	5: 22.1634
0	5: 22.1651
9	5: 22.1688
15	5: 22.1722
14	5: 22.1752
10	5: 22.1762
13	5: 22.1779
6	5: 22.1789

Notre Dame’s NEST Demonstrations

Authentication/Synchronization Demo

- Authentication Service estimates clock drift/offset to form secure neighborhoods
- Synchronization Service forces nodes in the network to “act” in synchrony
- Demonstration Script:
  - ✓ Start Network in AUTH state
  - ✓ Message traffic discussion
  - ✓ Use VisTool to visualize links
  - ✓ Identify clusters of sync’d nodes
- Demonstration Metric: Majority of Network Blinks LEDS in Synchrony

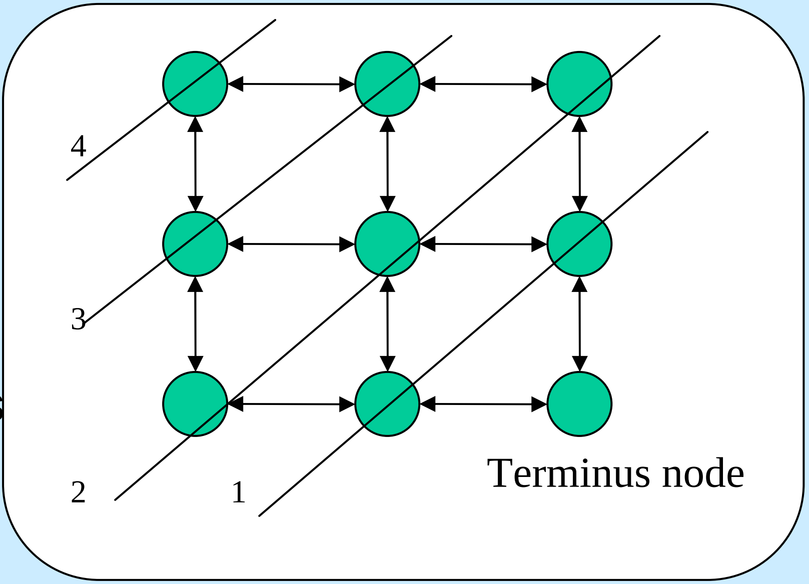
Discovery/Streaming Demo

- Discovery Service constructs min-hop routing table to specified terminus
- Streaming service transmits 100 packets at 1 packet per second.
- Demonstration Script:
  - ✓ All nodes start with all LEDs on
  - ✓ LEDs change to show min. hops to terminus
  - ✓ Start streaming and identify on-route nodes (counting LEDs)
  - ✓ Reduce power of on-route node and observe re-routing of connection
- Demonstration Metric: successful re-routing of connection

Route Discovery Service

Route Discovery Service

- Distributed Bellman-Ford minimum hop algorithm.
- Fixed terminus node whose periodic beacon generates sequenced DISC packets to prevent loops.
- Node’s neighborhood consists of nodes that it received DISC packets from.
- Node’s min-hop value is reset if next min-hop is not heard from.



Streaming Service

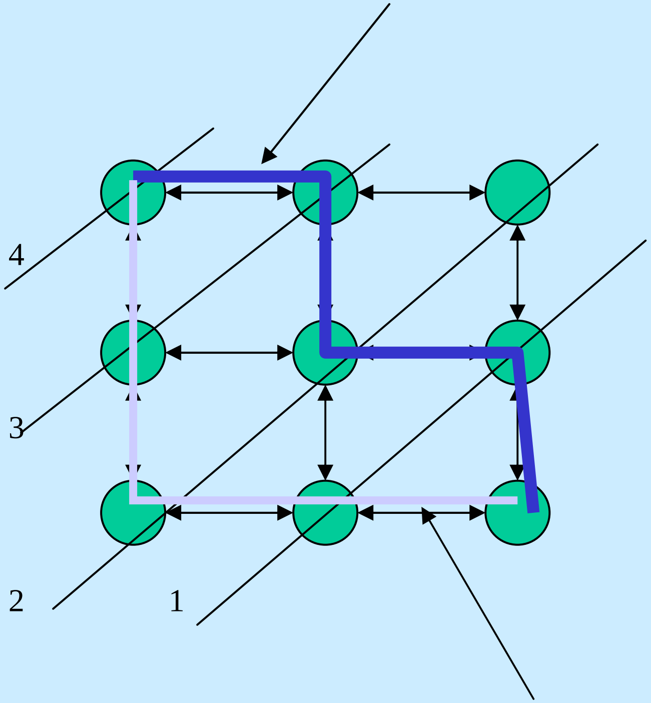
Streaming Service

- Route composed of next reliable min-hop.
- Adaptive streaming service estimates link reliability by listening to forwarded packets.
- Reliability estimates are reset if forwarded packets are never heard.
- Nodes that are not part of the route disable themselves.

Performance Metrics

- Total throughput versus power level
- Total number of strm packets
- Fast Re-routing of open connection

Primary route used for connection



Alternate route used by route reconfiguration service

Packet fields – Data Streaming Demo

Route Discovery Process:

7e 0 1f 7d 4c 35 0 0 42 2 0 0 2 0 ……

node ID seq. no. local cost

Streaming Process:

7e 0 29 7d 5b 1e 0 0 0 0 0 0 43 aa ff ff 43 aa ff ff ……

transmitting node seq. no. Streaming route (up t0 4 nodes)

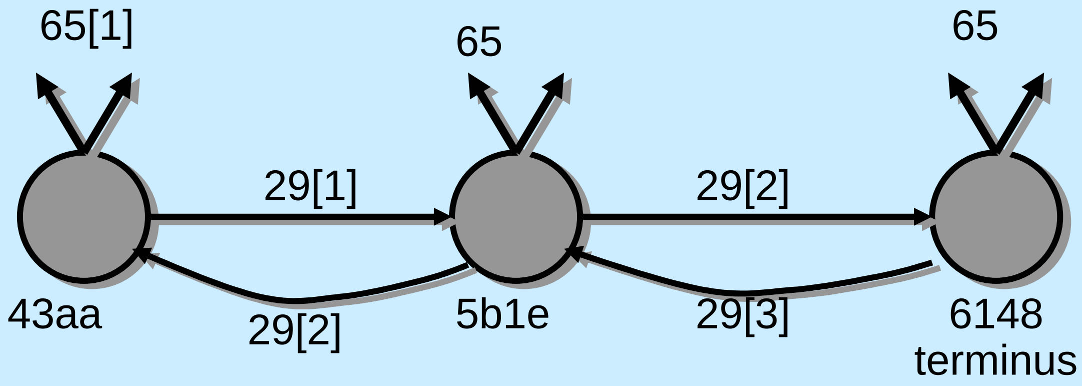
Throughput evaluation:

7e 0 65 7d 1a 0 0 0 63 0 0 0 43 aa ff ff …

packets relayed packets sent node ID

7e 0 65 7d 2e 0 0 0 34 0 0 0 5b 1e 0 0 ……

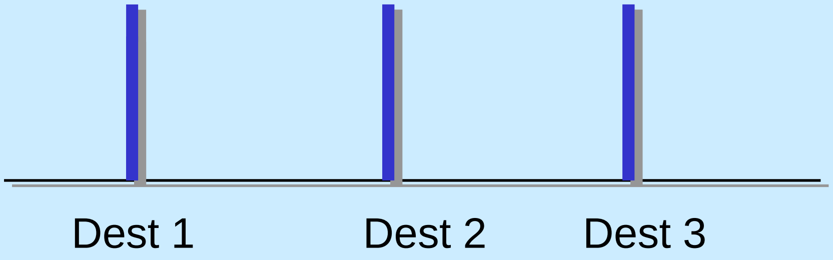
7e 0 65 7d 0 0 0 0 33 0 0 0 61 48 0 0 ……

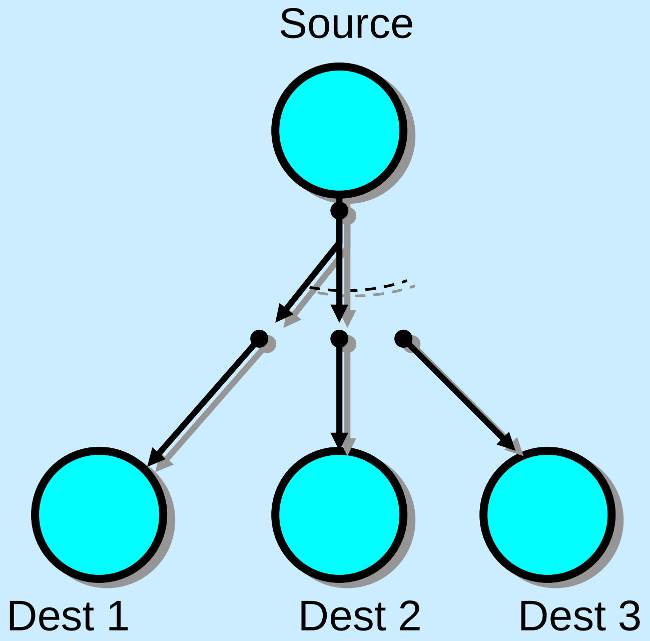


Link Reliability Estimation

- Problem: how to estimate link’s reliability if that link without sending a large number of test packets?
- Solution: set up a competitive process in which all next hop neighbors “inherit” the same initial reliability estimate.
- Experimental tests suggest that this approach leads to “fast” switching between alternative next hop nodes

Reliablity estimate of each dest. node





Relation to Challenge Problem

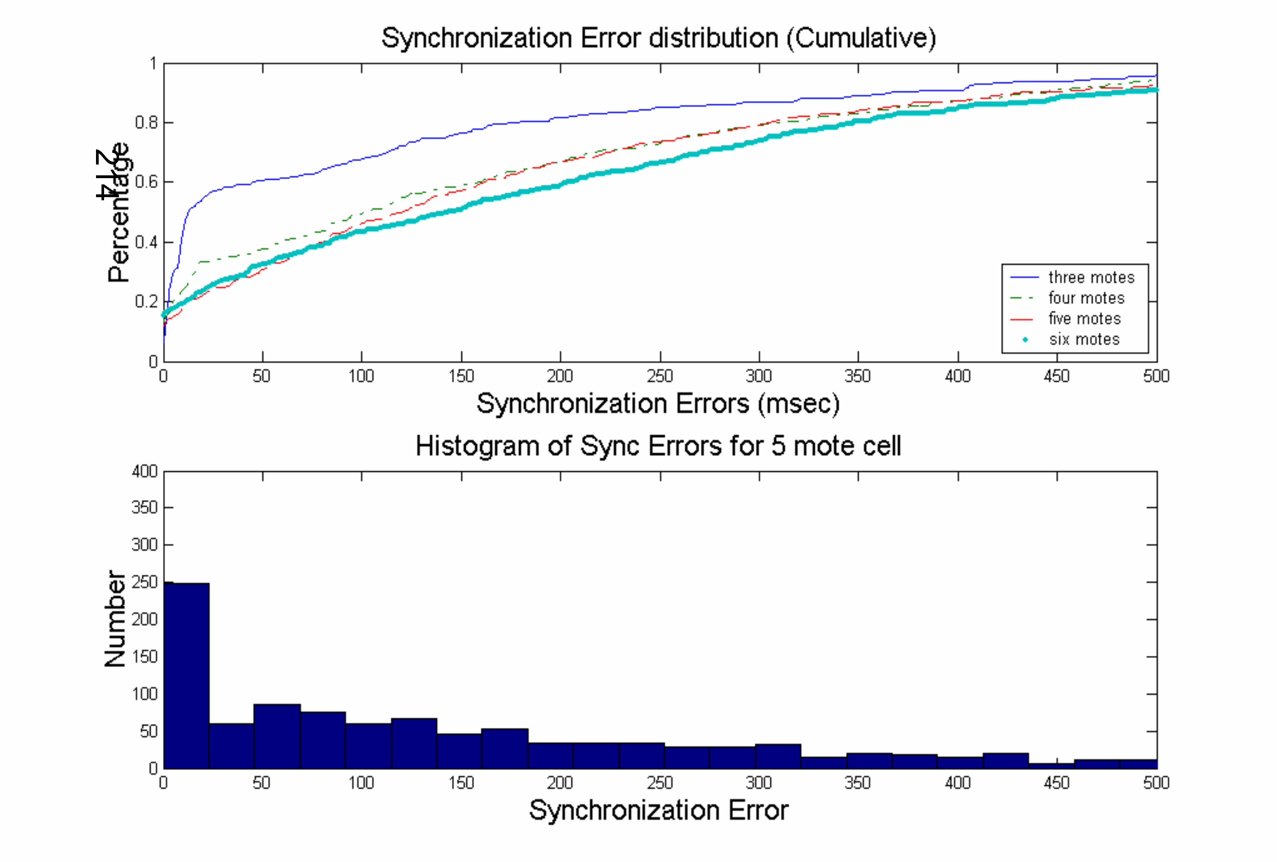
- **Authentication Service**
  - Easiest way to evade the pursuers is to disrupt their communication. Motes are an “open” platform that can be accessed by any mote sending the appropriate AM_msg. The authentication service secures motes from malicious users.
- **Synchronization Service**
  - In the challenge problem, motes will need to coordinate their actions in time. The clock synchronization service provides a common “heart beat” coordination within NEST.
- **Route Discovery Service**
  - The challenge problem may prefer content-based (rather than IP-based) addressing may be preferred. The route discovery algorithms used here provide a starting point for content-based routing.
- **Streaming Service**
  - The challenge problem’s NEST primary service is communication. One form of communication is streaming which will be useful in the challenge problem.
- **Telemetry Service**
  - The telemetry service presented here provides one possible method for an operator to monitor and supervise the behavior of the NEST. Without such a service it is impossible to evaluate the effectiveness of the challenge problem in an objective manner.
- **Coordination of NEST services**
  - NEST services must be coordinated with each other as well as user applications. The particular O/S services developed for this demo provide a starting point that can be used to coordinate multiple NEST and APP services in the challenge problem.

Service Evaluation Criteria

- Synchronization Service
  - Average clock synchronization error
  - Percent of network with synchronization errors below TBD msec
  - Synchronization errors in networks with drifting clocks.
- Dynamic Streaming Service
  - Throughput (received packets)
  - Message overhead required by dynamic streaming.
  - Re-routing in the presence of abrupt node/link failures

Synchronization Error

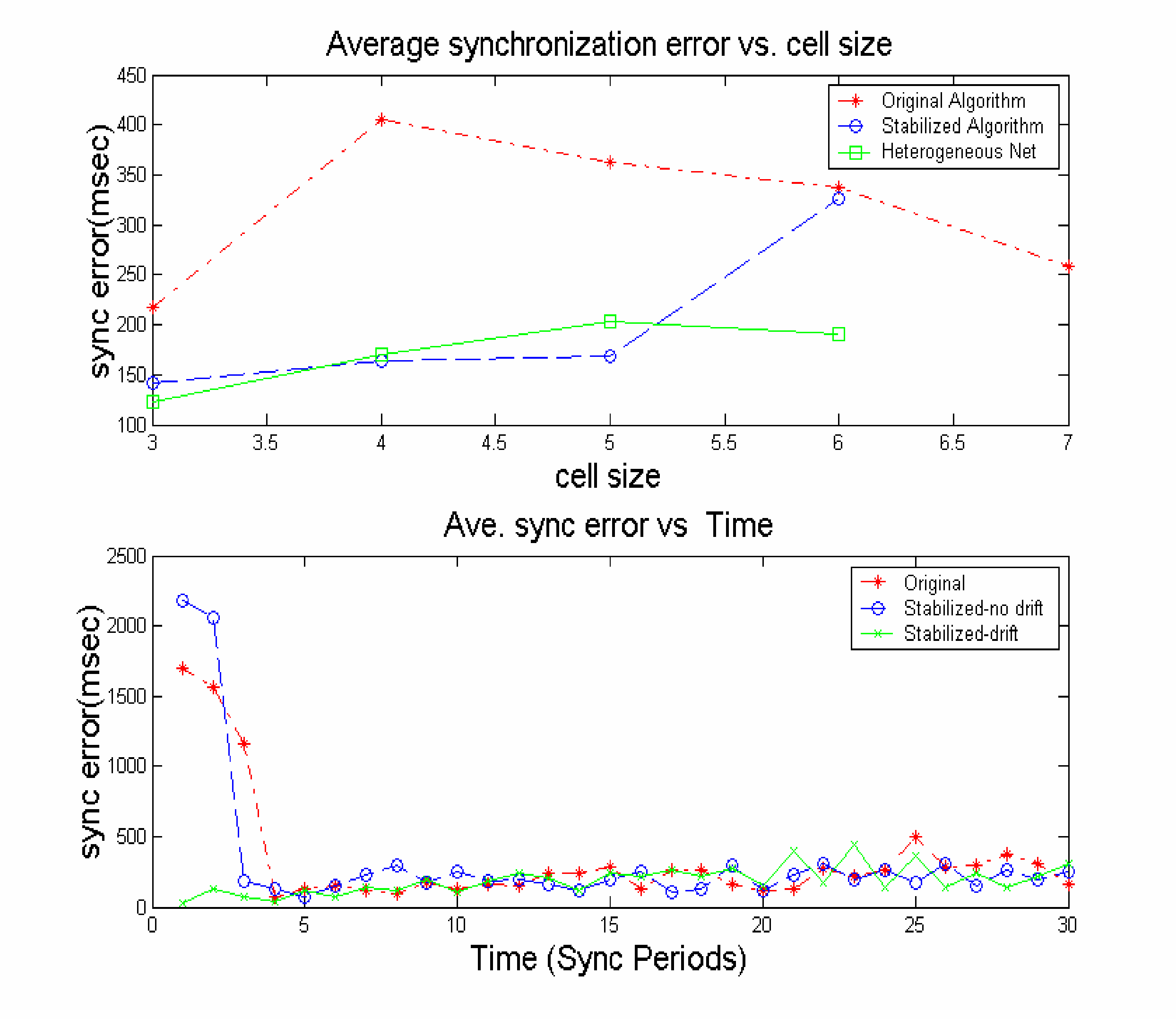
- 3-6 nodes in a single cell
- Start auth/sync services and wait till sync process settles down.
- 20-30 percent of cell has sync errors below 50 msec
- Fat tail of larger sync errors due to probabilistic drift/offset estimates



- Single cell containing a fixed number of nodes.
- Sync times measured with 1 msec accuracy.
- 10 runs lasting 30-100 sync epochs

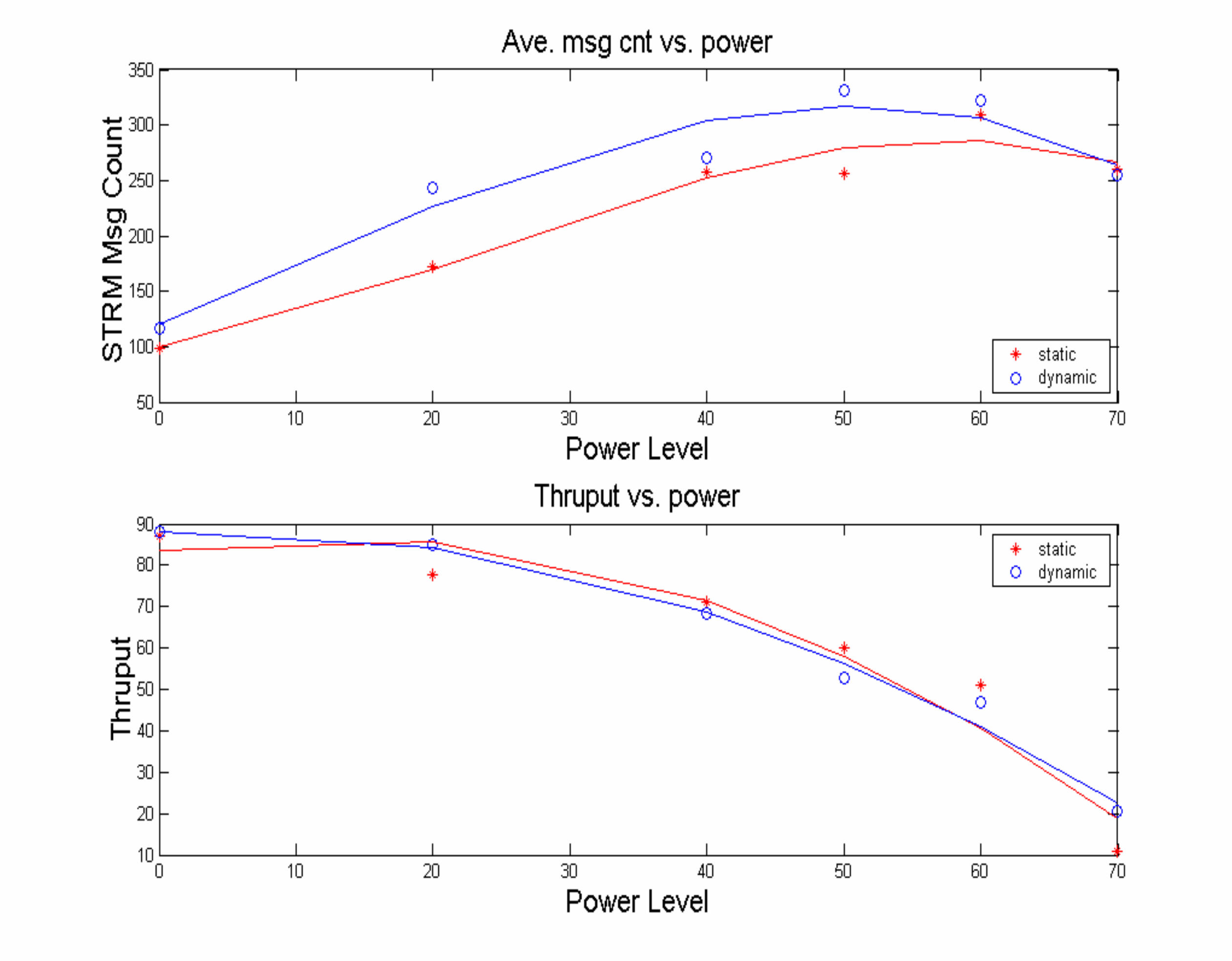
Heterogeneous Networks

- Instability in sync-alg. due to inconsistent drift estimates
- Stabilized sync algorithm improves sync err. by factor of 3.
- Sync process stabilizes after 5 sync epochs
- Nets with different clock drifts sync'd with 150 msec ave. error.



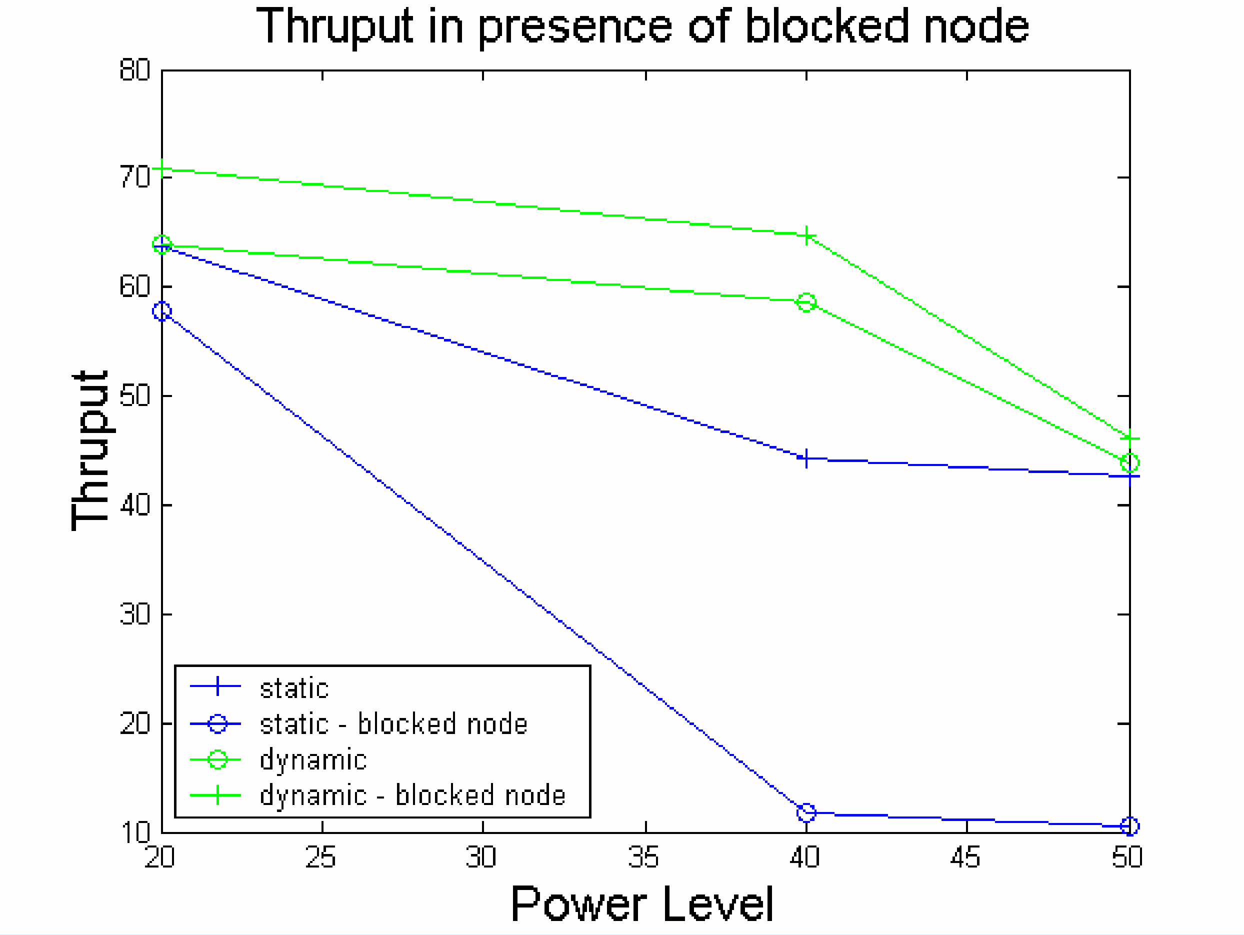
Streaming Performance

- Static and dynamic streaming have similar throughput (w/o node fault)
- Power usage grows linearly with decreasing power level until network becomes disconnected
- Dynamic streaming appears to have a constant message overhead that is interpreted as the cost required to “explore” alternate routes.



Adaptive Stream Reconfiguration

- **Procedure:**
  Start a streaming connection and then reduce power of on-route node
- **Static Routing**
  Blocking the route dramatically reduces connection thruput
- **Dynamic Routing**
  Network is able to discover alternate route and maintain connection's throughput.



Averages over 5 different runs.

Demonstration Issues

- Problem with original radio stack made it difficult to test routing/streaming algorithms until the end of April.
- Berkeley's radio stack uses nearly all of the OC/IC interrupts. We needed to rewrite parts of the stack to free up interrupts for driving robotic platform.
- We perceive the selected RF transceiver as a point of weakness in the MICA/RENE. We're currently designing a spread-spectrum comm. module that interfaces to the MICA.
- MICA sensor board uses every available I/O pin on the micro-controller. We've been using the Rene sensor boards to develop custom sensor boards.
- Programming board and Rene sensor boards from Xbow are ridiculously expensive.
- Relatively slow connect speeds make the MICAs difficult to use for real-time control.

Lessons Learned

- Probabilistic nature of the Berkeley OEP dramatically effects the behavior of distributed algorithms
  - Clustering in sync algorithm
  - Instability of original sync algorithm
- Reconfiguration of network services requires novel methods for making switching decisions that are fast and take into account the probabilistic nature of the channel.
- NEST services will need to be “cross-layer” in that high-level network services may dramatically improve performance by controlling or measuring physical layer variables.
- Berkeley OEP is susceptible to denial-of-service attacks and this should motivate the use of more sophisticated spread-spectrum RF communication modules. NEST services can extra channel diversity to improve middleware performance.

Real-time Configuration of Networked Embedded Systems

P.J. Antsaklis (PI), Martin Haenggi, M.D. Lemmon
Dept. of Electrical Engineering
University of Notre Dame

DAPRA/IXO NEST PI MEETING
July 2003, San Francisco CA

Subcontractors and Collaborators

Senior Personnel

Panos J. Antsaklis (PI), Michael D. Lemmon, Martin Haenggi

Electrical Engineering, University of Notre Dame

Graduate Students

Luis Montestruque, Yashan Sun, Hui Fang, Marian Iordache, Lei Fang, Ioannis Koutroulis, Brett McMickell, Min Xie

Real-time Reconfiguration of Networked Embedded Systems

University of Notre Dame

Contract No. F30602-01-2-0526

AO No: L546/00

Period: 5/2001 – 9/2004

Panos J. Antsaklis (PI)

Department of Electrical Engineering

PI Phone No: 574-631-5792, Fax 574-631-4393, antsaklis.1@nd.edu

DARPA Program Manager: Vijay Raghavan

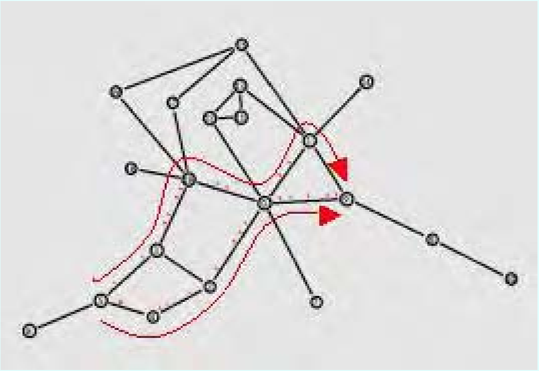
Agent name & organization: Juan Carbonell, AFRL/Rome

Univ. of Notre Dame

Real-time Reconfiguration of Networked Embedded Systems

Problem and Challenge

Real-time reconfiguration of middleware services in networks of embedded systems



Real-time+ Distributed Algorithms + Feedback Loops+Probabilistic Analysis+Formal Verification

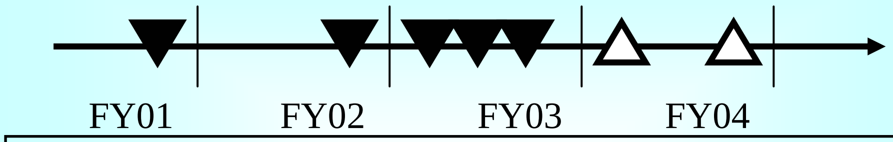
New Ideas

- Model-based clock synchronization and security for heterogeneous networks of embedded systems
- Adaptive reconfiguration of middleware services in response to abrupt network changes
- Real-time network management through probabilistic analysis and formal verification.
- Exploiting feedback in distributed algorithms

Impact

- Middleware for ad hoc networks of embedded systems.
- Real-time reconfiguration of streaming connections in ad hoc wireless networks.
- Scalable methods for formation control in robotic colonies

Schedule



3QFY01: Project Start
3QFY02: Berkeley OEP Demo
1QFY03: Demo of Model Problem (robot formation control)
2QFY03: Delivery of TInyOS Service Components
3QFY03: Integration with Group I components
1QFY04: Challenge Problem Integration
3QFY04: Final Demonstration

Problem Description and Objectives

Adaptive reconfiguration in fine grained networks of embedded systems that interact tightly with the physical world so to maintain Quality of Service (QoS.)

Project Tasks:

- Clock-synchronization:
  Develop clock-sync algorithms for ad hoc networks.

- Adaptive Reconfiguration
  Fast reconfiguration of NEST services in the presence of abrupt network changes

- OEP Integration
  Red Force Tagging for SOCOM demo

Project Status

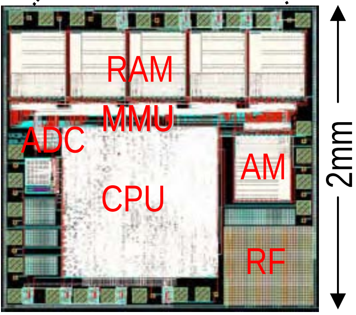
PROGRESS SINCE LAST MTG.

- Red Force Tagging Demo
  - design and built TagMote
  - NesC components to control TagMote
  - Modified NEST Services to support Demo
  - RedForceTagging Trajectory Reconstruction algorithms
- NEST Middleware Services
  - NesC/MICA2 version of TimeSync.
  - Reliable Data Download Service (NesC/MICA2)
- Wireless Sensor Networks
  - Amplifier Power effect on connectivity and routing
- Overload Management in Sensor Actuator Networks

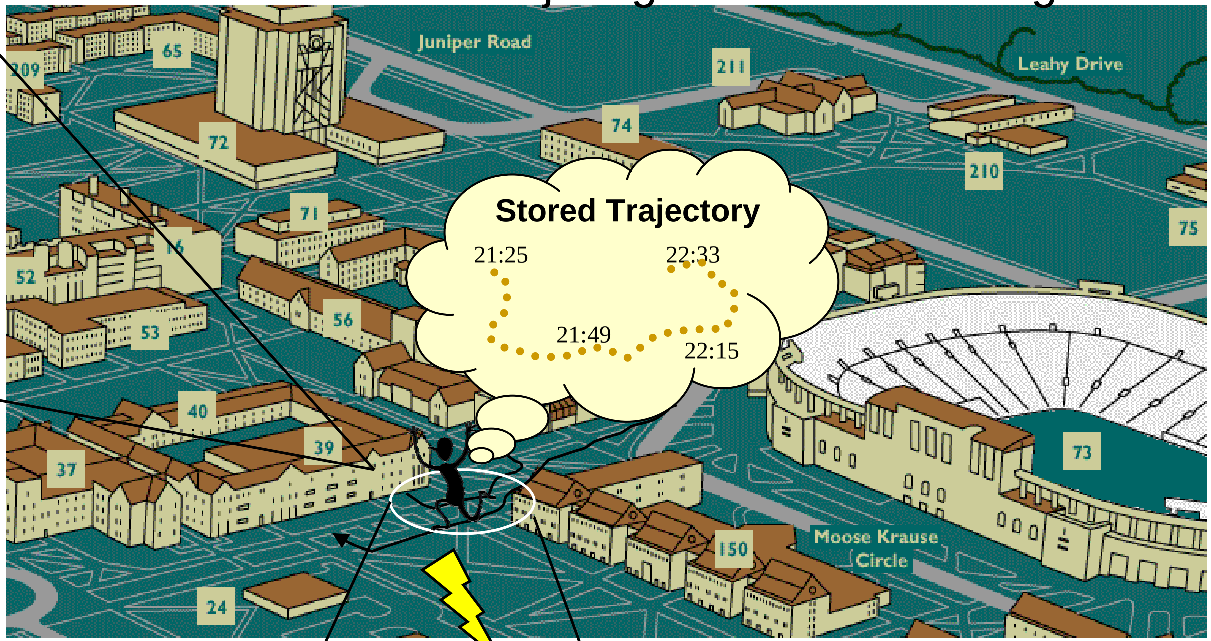
Outline

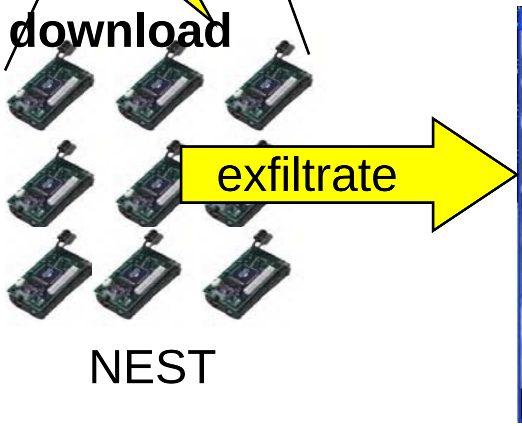
- **Red Force Tagging**
  - designed and built a sensor network with a “mobile” TagMote that logs sensor data when away from the NEST, downloads data upon returning to the NEST, and then exfiltrates data from NEST to RelayMote.
- Wireless Sensor Networks
  - Impact of Rayleigh fading on NEST middleware services
- Overload Management in Sensor/Actuator Networks
  - Relationship between application performance and network Quality-of-Service.
- Program Specifics
  - Milestones and future plans

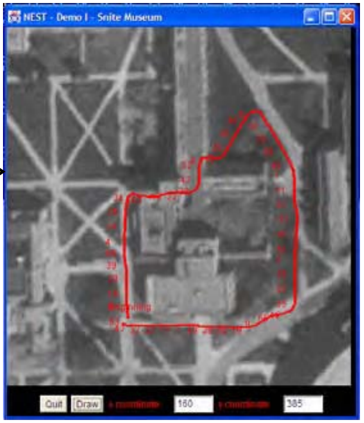
Red Force Tagging Concept of Operations



Where did the subject go after the meeting?







Tag subject at meeting with device that records his path after the meeting.

Download/exfiltrate stored trajectory data using NEST middleware services

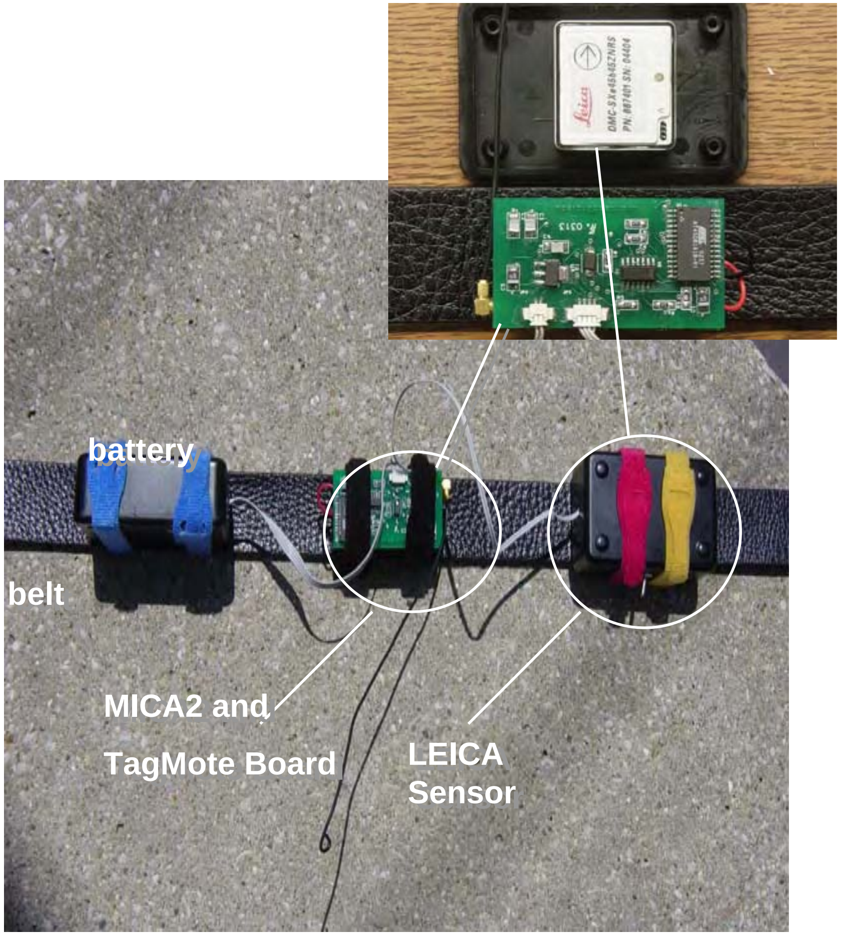
Relation to Other NEST Projects

- Common with Other NEST Projects
  - –Uses NEST services for network synchronization, localization, routing
  - –The mote network could be the sensor network used for the other NEST/SOCOM projects

- Different from Other NEST Projects
  - –TagMote. Uses a special mobile sensor mote that collects data which are automatically downloaded to the mote network when in proximity
  - –The mote network is used as a communication network not as a sensor network, so different density requirements
  - –Routing of different class of data. Large amounts of data from many sources, typically the motes at the edges of the network, to single destination, the Relay mote.

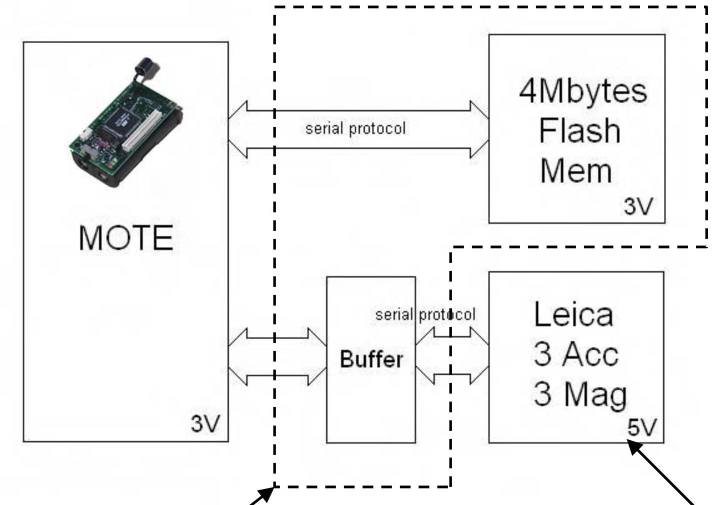
TagMote

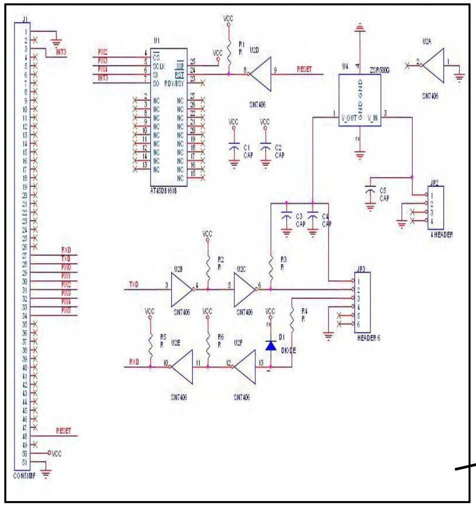
- **Design and Build a special “TagMote”**
  - MICA2 for communication with the NEST
  - Leica Sensor
    3D-accelerometer
    magnetic compass
  - Specially designed TagMote board with 4 MByte Flash Memory
  - currently attached to a belt that is worn by the subject

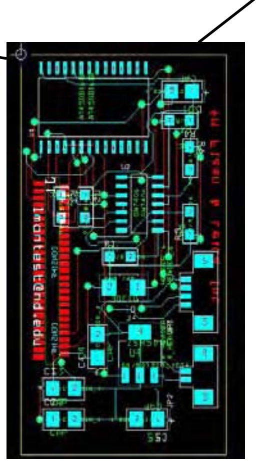


TagMote Hardware Overview

- The TagMote incorporates a Berkeley mote to provide processing power, temporary storage and communications to the network.
- The TagMote uses a 3D-accelerometer and a 3D-digital magnetic compass (Leica) to collect data regarding distance traveled and direction, and it uses a flash memory to store the data.
- TagMote Board:
  4-Mbytes Flash
  Buffer
  TinyOS components to manage Leica Module





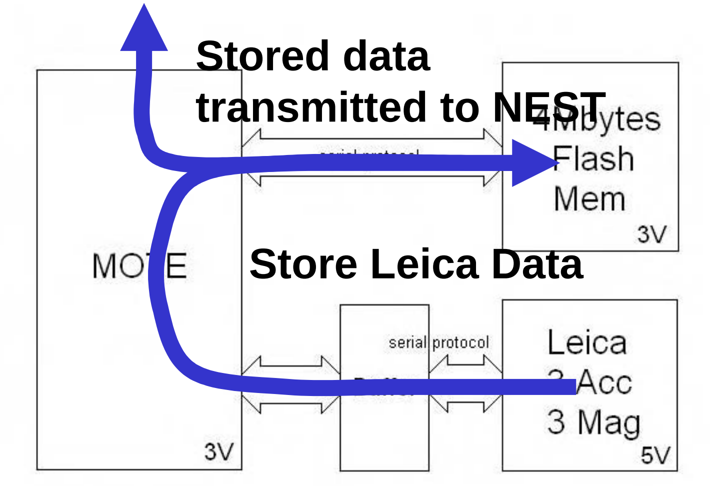


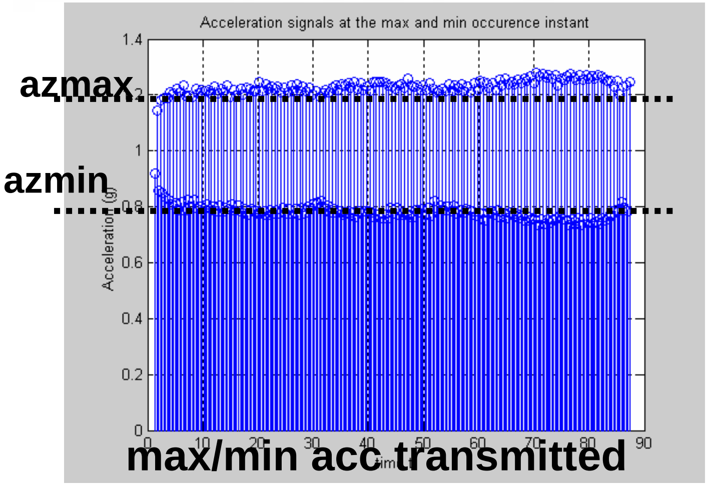
TagMote Board

Leica Module

TagMote - On Board Processing

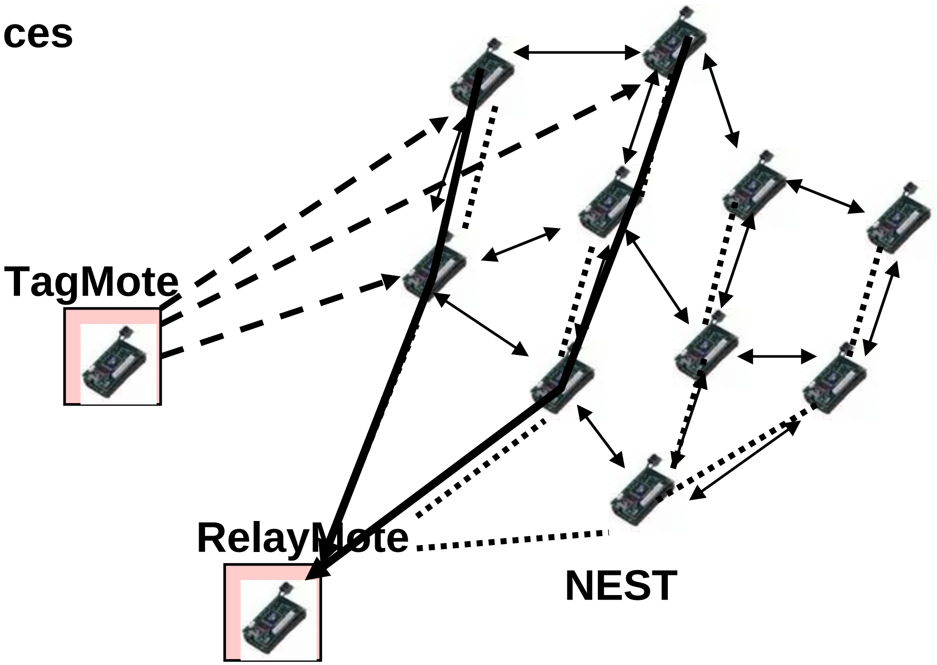
- **Sensor data (Mag. & Acc.) stored in 4Mbytes Flash (360 bytes/sec)**
  - Circular buffer structures and task driven write and read processes minimize blocking of other interrupt driven processes.
- **Data Processing when Tagmote is transmitting.**
  - FIFO buffer stores last 8 samples as data comes out.
  - Data of FIFO is analyzed for detection of inflexion points (max. or min. detection).
  - If an inflexion point is detected then data over the buffer is averaged and transmitted (for all 6 measurements.)
  - Transmission of max/min acceleration greatly reduces the amount of data that needs to be downloaded to the network





Technical Approach – NEST Middleware

- **Design and Build NEST Middleware Services to**
  - automatically download of large amounts of data from TagMote to NEST
  - automatically stream NEST data to SOCOM relay
  - autonomously initialize and maintain NEST services required to support the Red Force Tagging mission.
- **Red Force Tagging Middleware Services**
  - Clock Synchronization
  - Ping/Dump (tagmote to NEST)
  - Backbone construction
  - Streaming from NEST to Relay
  - TLM service to monitor NEST

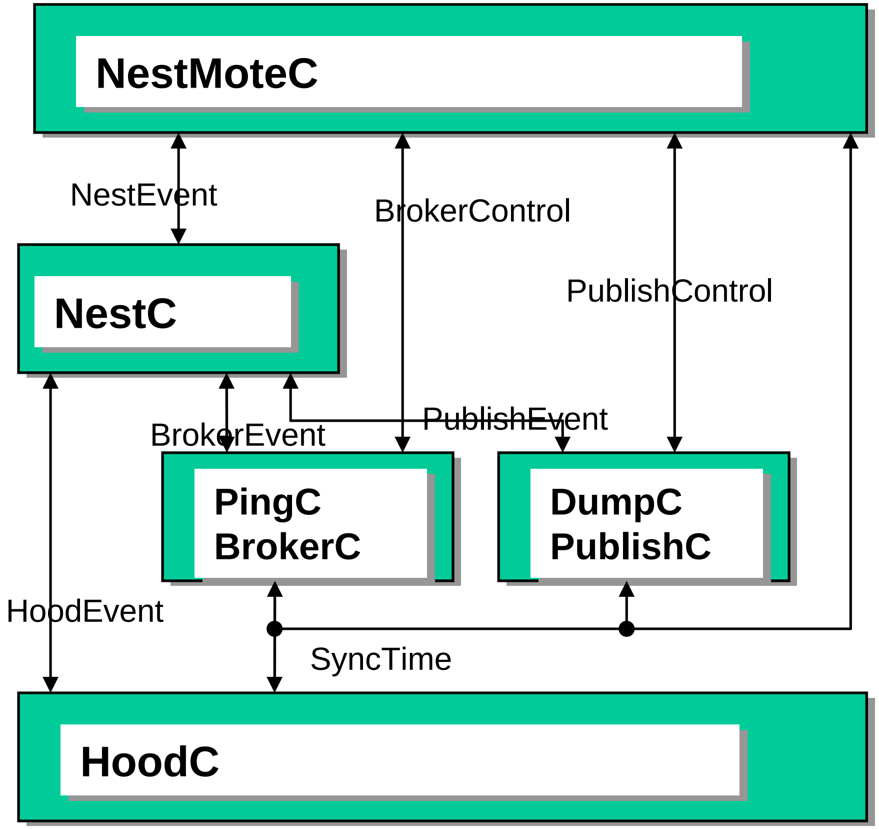


RedForceTagging Middleware Architecture

NEST Middleware provides the “glue” that enables the TagMote and NestMotes to interact as a single system

NEST Middleware allows the “flexibility” of only downloading when the resources are available.

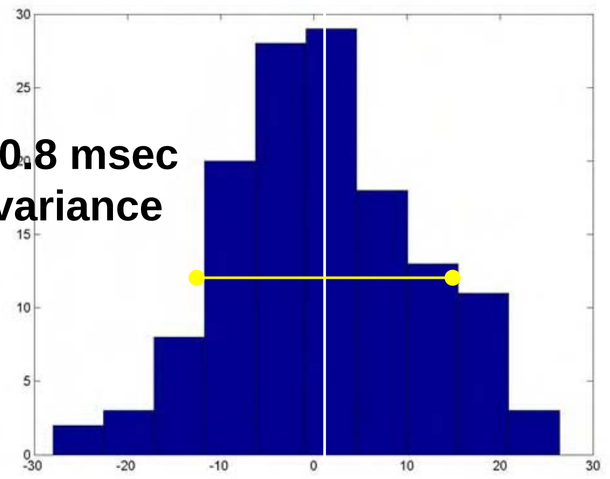
- Shared NEST Services
  - – Neighborhood initialization
    build tables of neighborhood ID’s
  - – Global clock synchronization
    maintenance of global network clock tick
  - – Localization (not implemented)
    estimate global position of TagMote
- RedForceTagging Services
  - – Ping Service
    used by TagMote to find a NestMote
  - – Data Dump Service
    used by TagMote to download (dump)
    data to NestMote EEPROM
  - – Broker Service
    builds spanning tree back to RelayMote
  - – Publication Service
    routes dumped TagMote data to RelayMote



RedForceTagging Middleware Architecture is a set of interacting components.

Clock Synchronization Service

- **Clock Sync Service maintains a “global” clock tick throughout the entire network.**
  - useful in coordinating node actions
  - useful in backbone (route) generation algorithms
  - based on local broadcast mechanism
- **Current Implementation**
  - TinyOS component (HoodC)
  - Very robust operation
  - 10 msec variance in event synchronization due to jitter in update.
  - potential 20 micro-sec sync possible which enables TDMA communication



mean = 1.8 msec

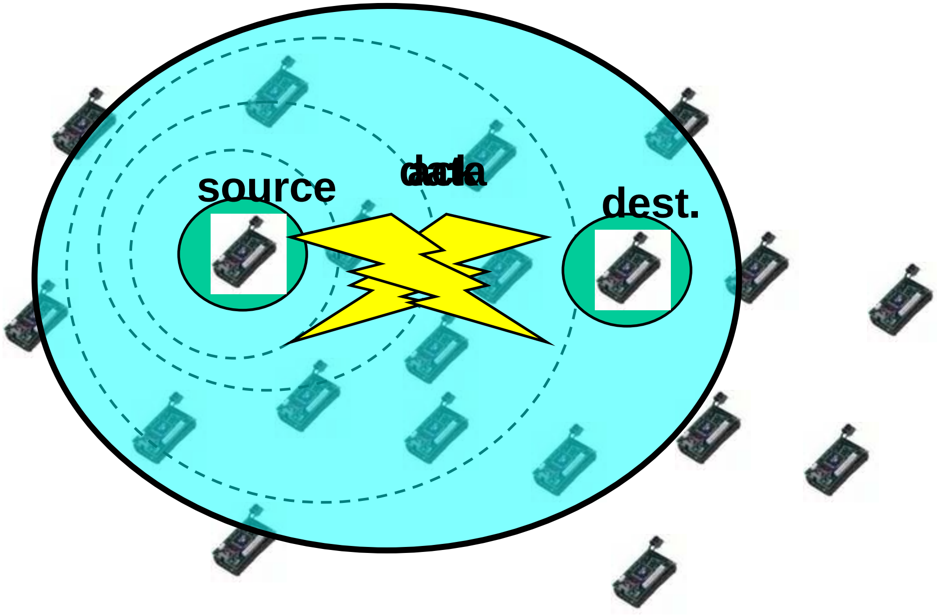
10.8 msec variance

Histogram of Sync Jitter over 6 hour period (50 node network)

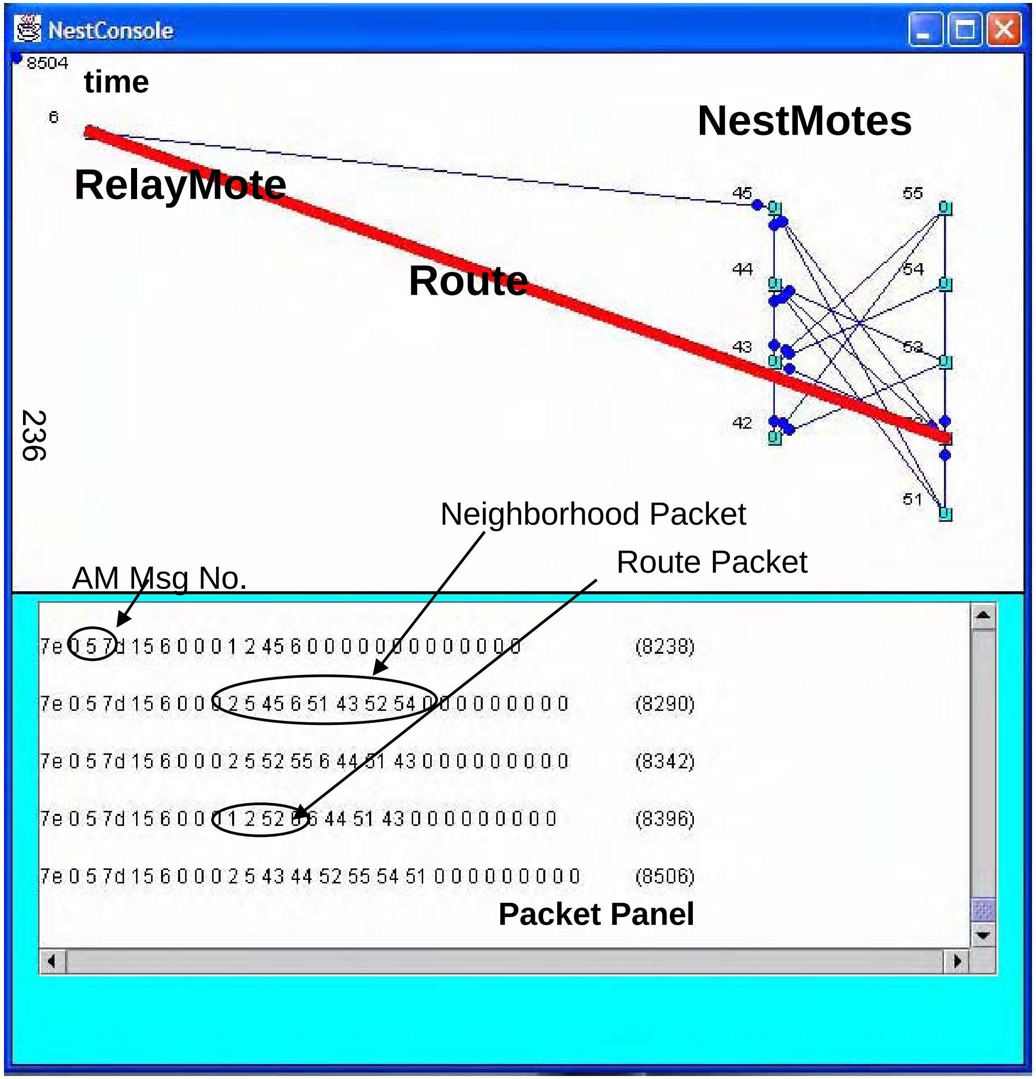
Download and Streaming Services

- **The traffic generated by this application “streams” a large amount of data in a relatively short period of time.**
  - – Downloads must be “reliable”
  - – Downloaded data must be “logged” to flash EEPROM 16 bytes / 10 msec
  - – Preprocessing of data reduces speed at which download can be done.
  - – Achievable download rates
    data for 3-minute walk downloaded to NEST in 3-minutes

- **Streaming service protocol**
  - – Control packets set up a Basic Service Set (cell)
  - – All transmitted data packets are acknowledged
  - – Cell is released if any data/ack packet is lost.



Telemetry Service



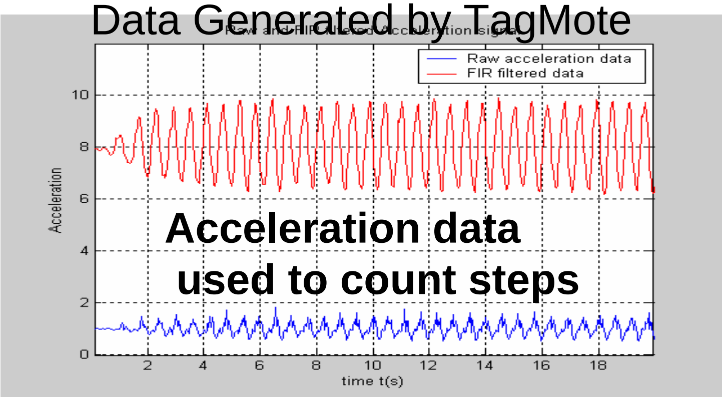
```
7e 0 5 7d 15 6 0 0 0 1 2 45 6 0 0 0 0 0 0 0 0 0 0 0 0 0      (8238)
7e 0 5 7d 15 6 0 0 0 2 5 45 6 51 43 52 54 0 0 0 0 0 0 0 0 0  (8290)
7e 0 5 7d 15 6 0 0 0 2 5 52 55 6 44 51 43 0 0 0 0 0 0 0 0 0  (8342)
7e 0 5 7d 15 6 0 0 0 1 2 52 6 6 44 51 43 0 0 0 0 0 0 0 0 0   (8396)
7e 0 5 7d 15 6 0 0 0 2 5 43 44 52 55 54 51 0 0 0 0 0 0 0 0 0 (8506)
```

- after backbone built, TLM service started
- AM msg 5
  - Neighborhood packets
  - Route packets
- Relay uses TLM service to build a graphical view of the network
- Mote SerialID's encode location

Dead Reckoning and Trajectory Reconstruction

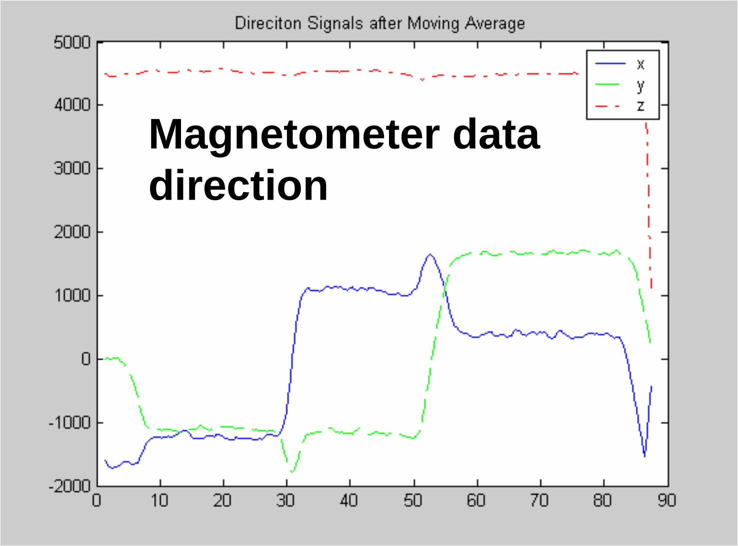
Trajectory Reconstruction Accomplished using Dead Reckoning

Data Generated by TagMote

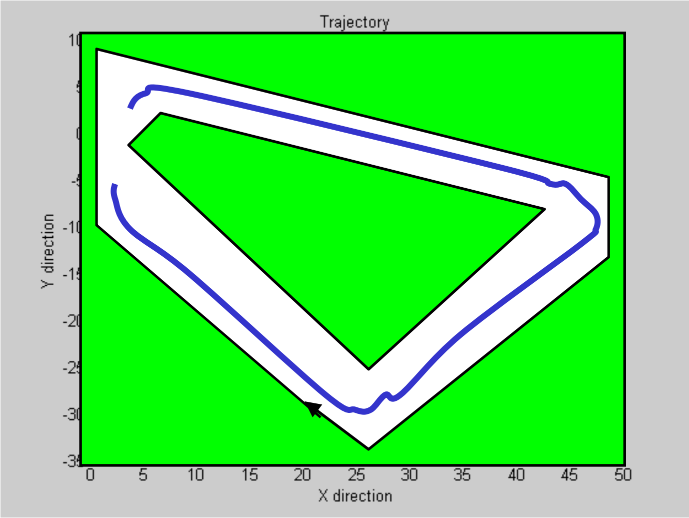


Acceleration data used to count steps

- Count steps, estimate step length
- Subject’s direction determined from magnetometer
- Trajectory reconstructed at base station from step count, step length, and direction
- Part of processing done on TagMote Bulk of trajectory reconstruction done at RelayMote

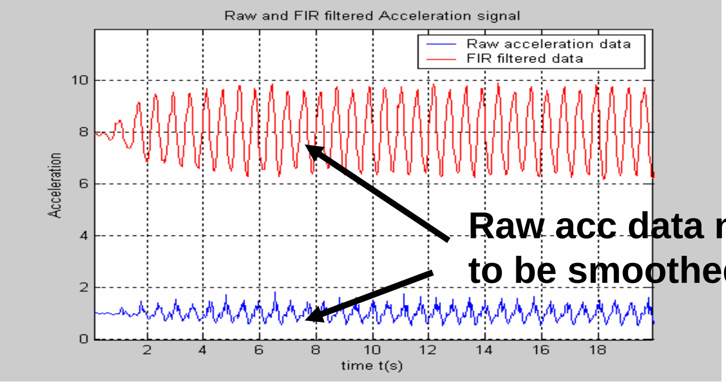


Magnetometer data direction

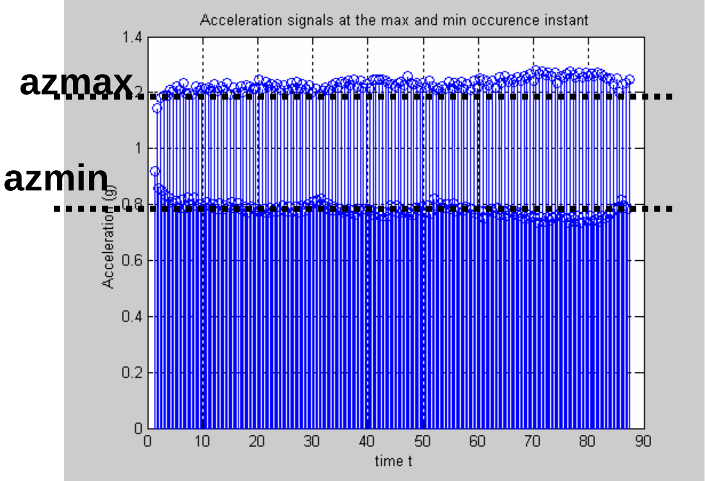


Trajectory Reconstructed at RelayMote

Step Length Estimation



Raw acc data needs to be smoothed

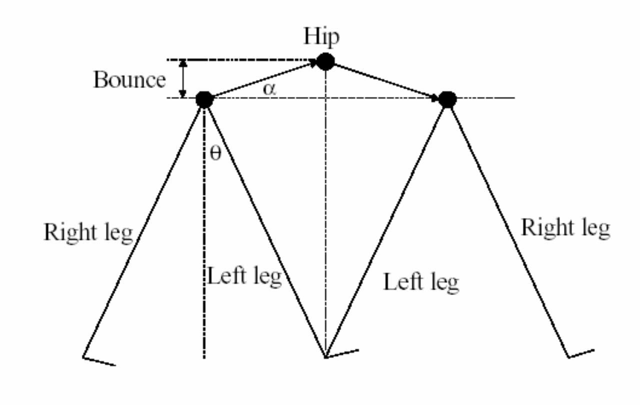


- Raw acceleration data is smoothed to more clearly identify peaks in acceleration profile due to steps

- Min and max accerations are identified in order to estimate step length.

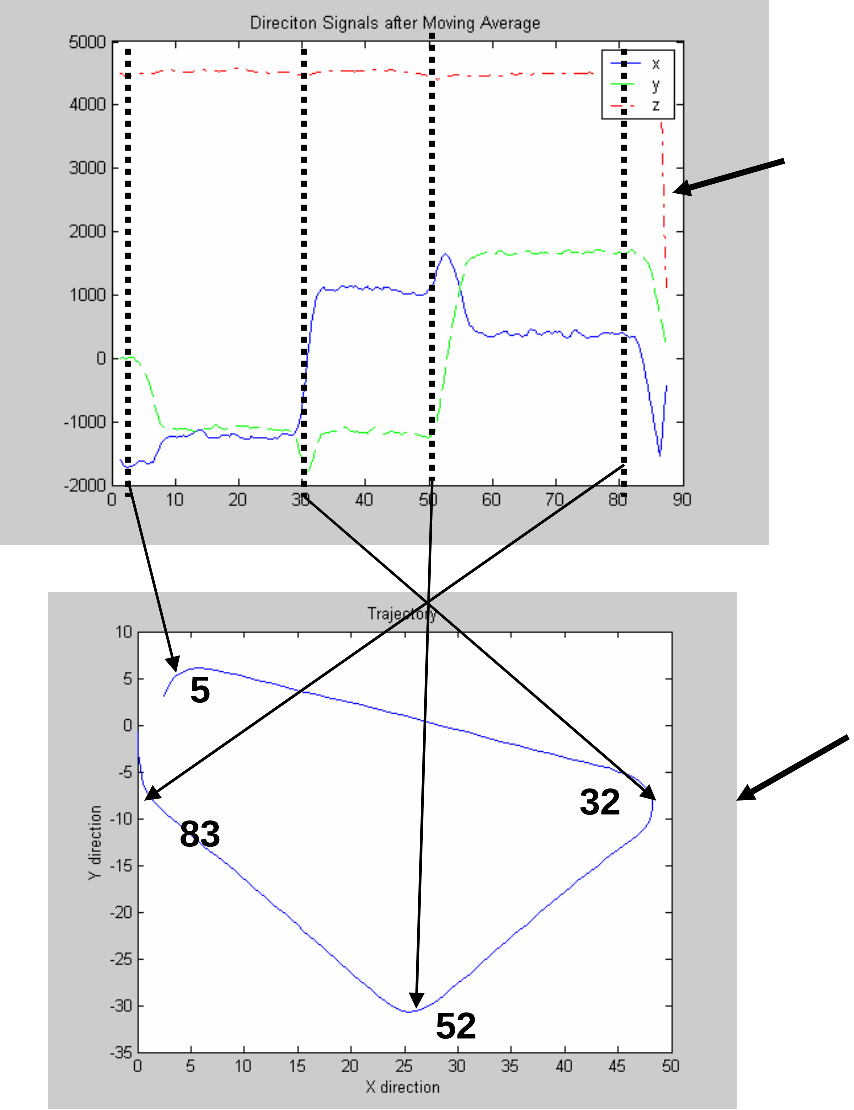
- Step length estimated according to formula

$$K \times \sqrt[4]{a_{z\max} - a_{z\min}}$$



This processing can be done on TagMote

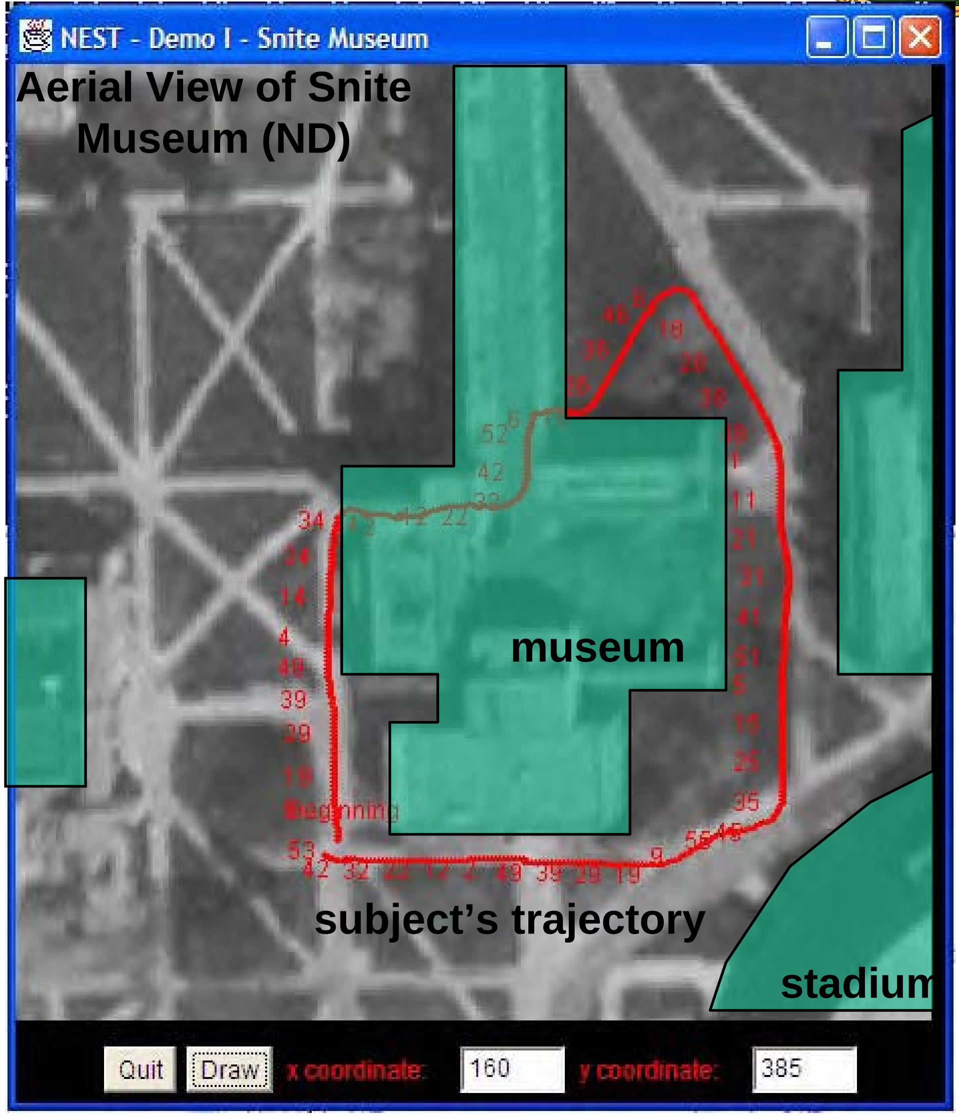
Trajectory Reconstruction



- Direction measurements must be filtered to remove periodic disturbance due to subject’s gait.
- Distance traveled in a given direction is obtained by multiplying step length by number of steps.
- Entire Trajectory is reconstructed by piecing the segments together and then displayed against a “map” of the region.

Trajectory GUI

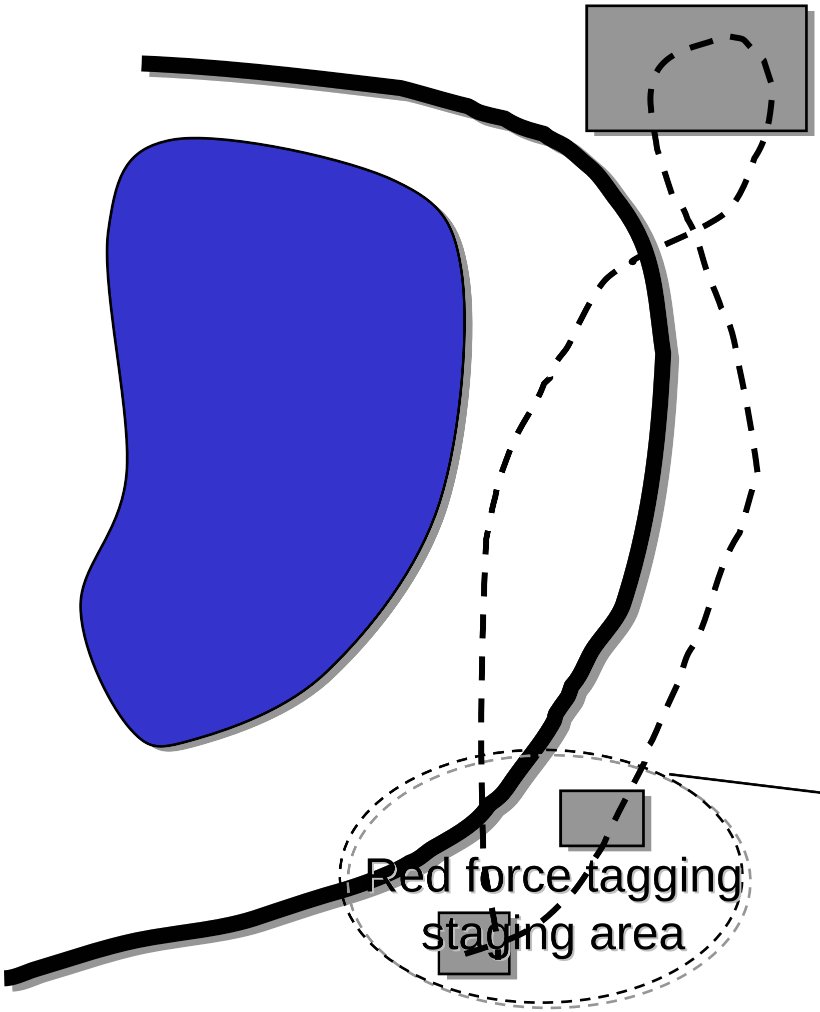
- Data captured by NestConsole GUI is written to a file
- This data is processed by a Java program to generate the subject's trajectory
- The trajectory data is then plotted against an aerial view of the area
- This particular plot shows trajectory reconstruction in a GPS denied area (within bldg).


NEST - Demo I - Snite Museum
Aerial View of Snite Museum (ND)
museum
subject's trajectory
stadium
Beginning
Quit
Draw

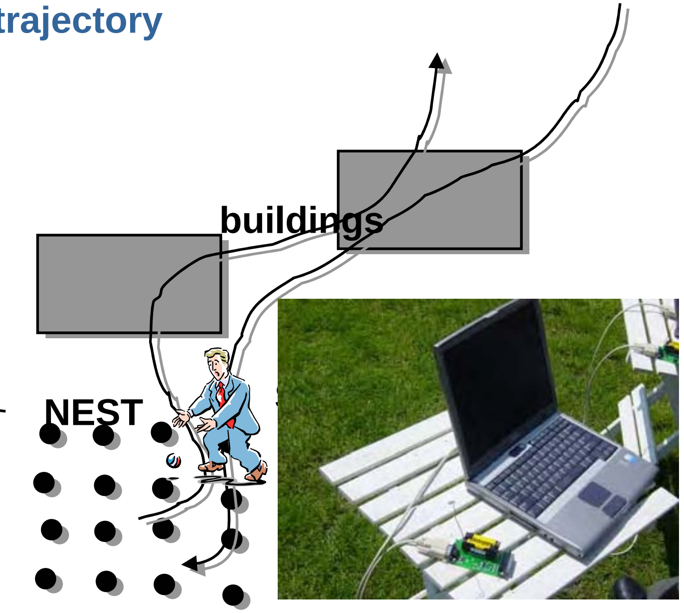
x coordinate: 160 y coordinate: 385

Red Force Tagging Physical Layout

- **Physical layout of RedForce Tagging Staging Area**



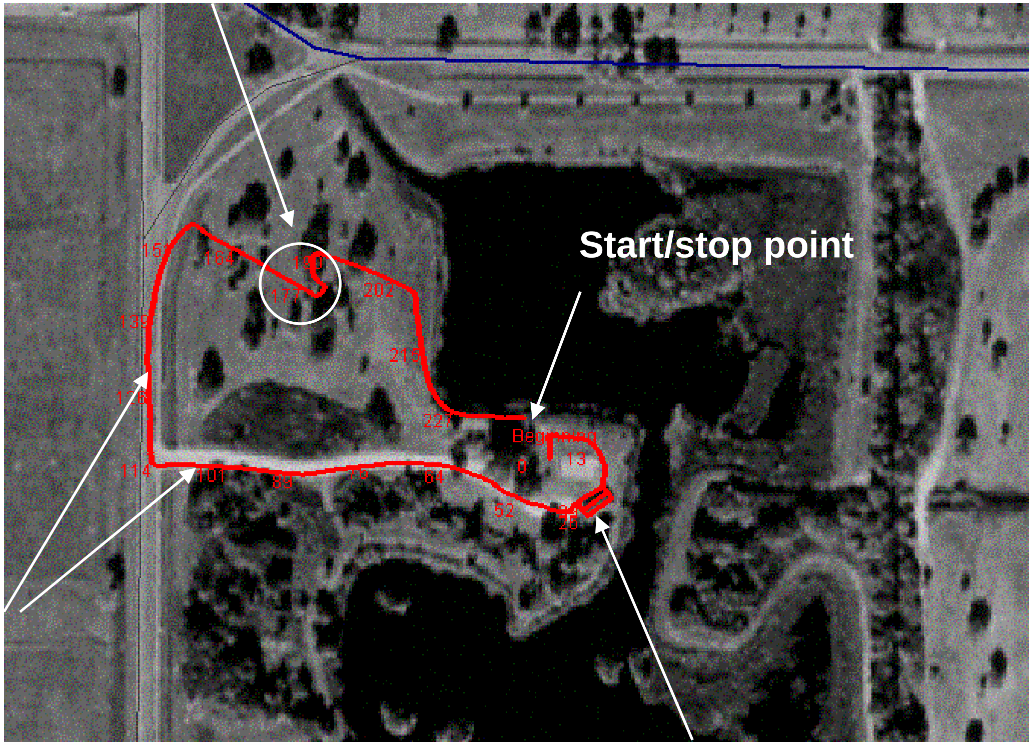
1. **Subject attaches TagMote Belt and moves to initial starting position**
2. **Subject takes a 3-5 minute walk through the test course**
3. **Subject returns and data downloaded to NestMotes**
4. **Listener displays reconstructed trajectory**
5. **Relay displays reconstructed trajectory**



Performance at Tampa Demo

- 5 meter error over 800 meter walk
  20 meter over 3 kilometer walk
- exfiltration and download time
  - 1 minute of real-time to 1 minute of download time.
- exfiltrated trajectory example from Lewis Lake
- Works in GPS denied areas

Maneuver to avoid water hazard



Start/stop point

Following road

Maneuver through bldg

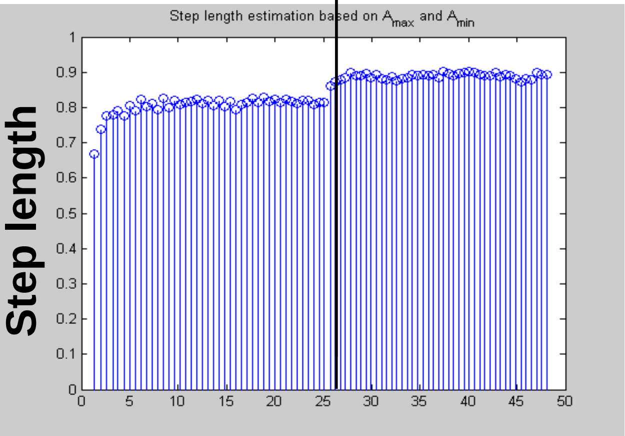
Gait Analysis

Acceleration and step length histories can be used to analyze subject's gait

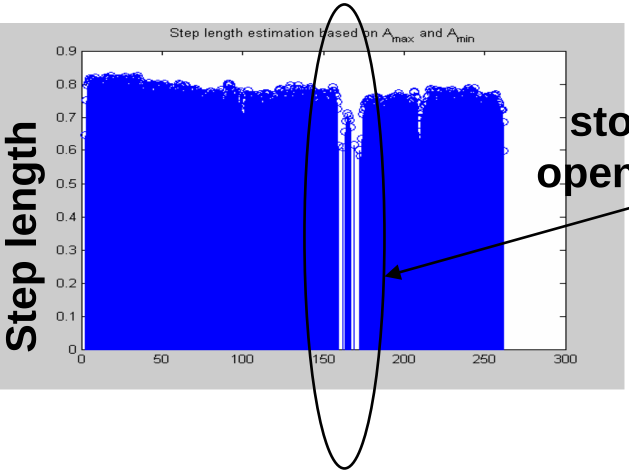
- **gait changes walking/running**
- **stopping to open doors**

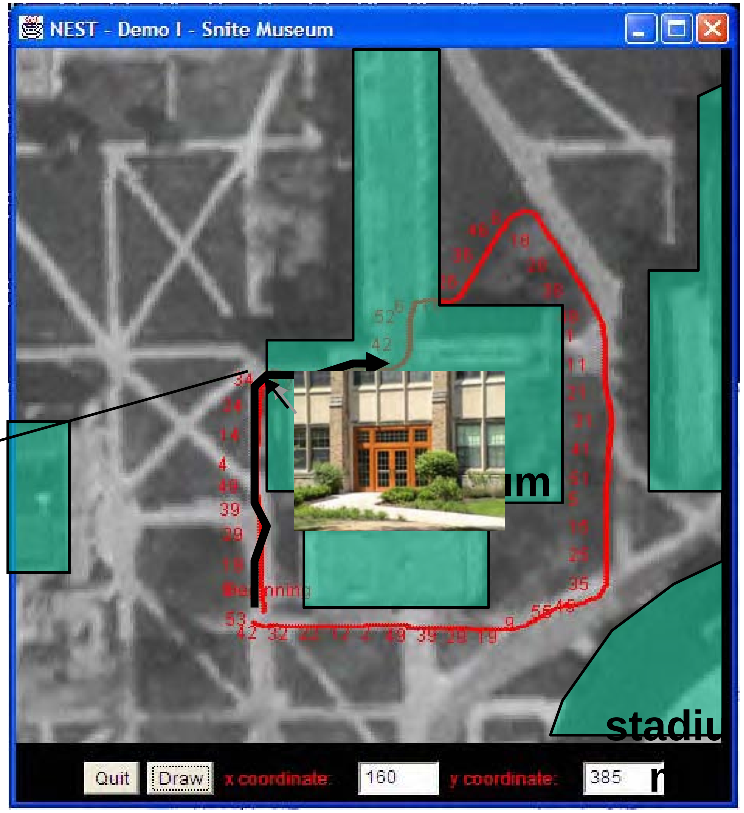
ange

change in stride length

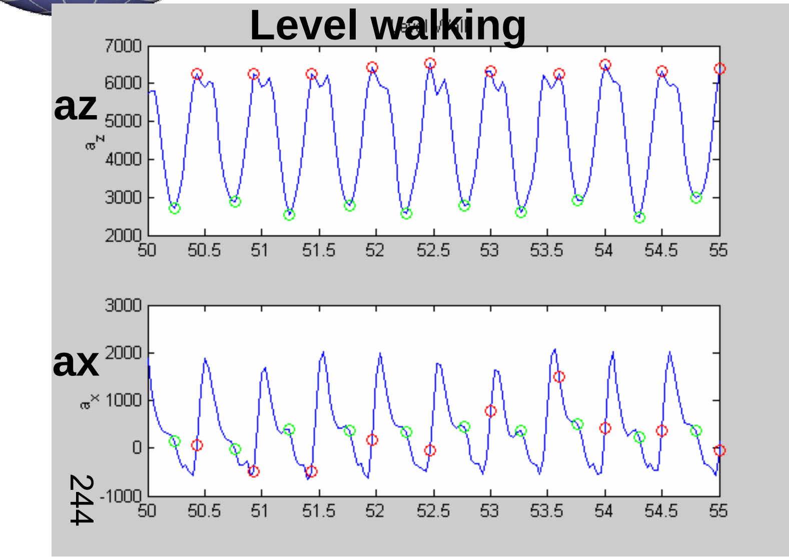


stop to open door

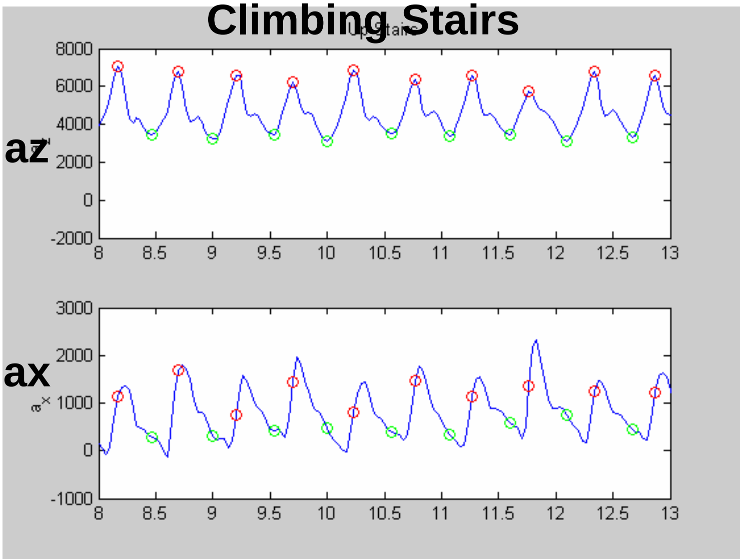




Gait Analysis



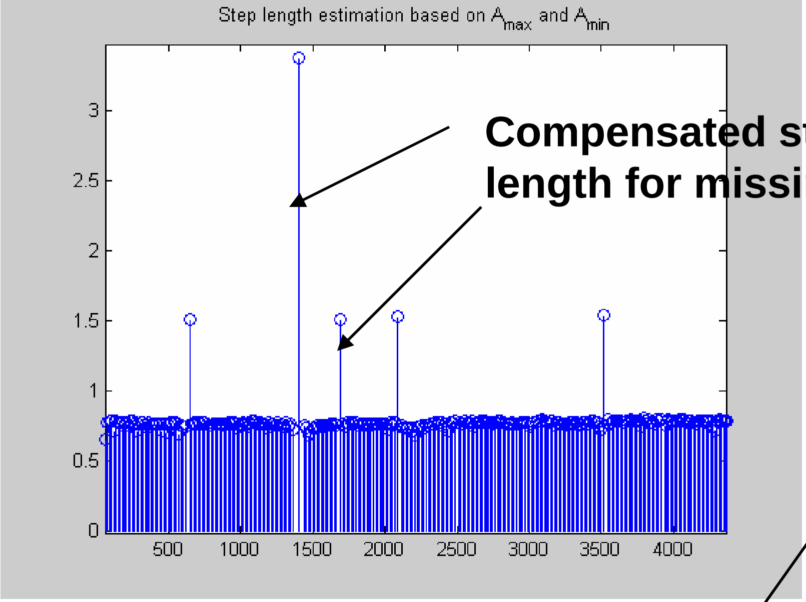
Level walking



Climbing Stairs

- It is possible to analyze acceleration profiles to gather additional information about the subject
- It is possible to discriminate between level walk or climbing stairs from max/min acceleration profiles.
  - Step length changes when climbing stairs
  - enhances position estimation accuracy
  - barometers aren’t always dependable, but smoothing barometer information with sgait discrimination allows more accurate 3-d trajectory reconstruction within buildings

Compensation for Missing Data

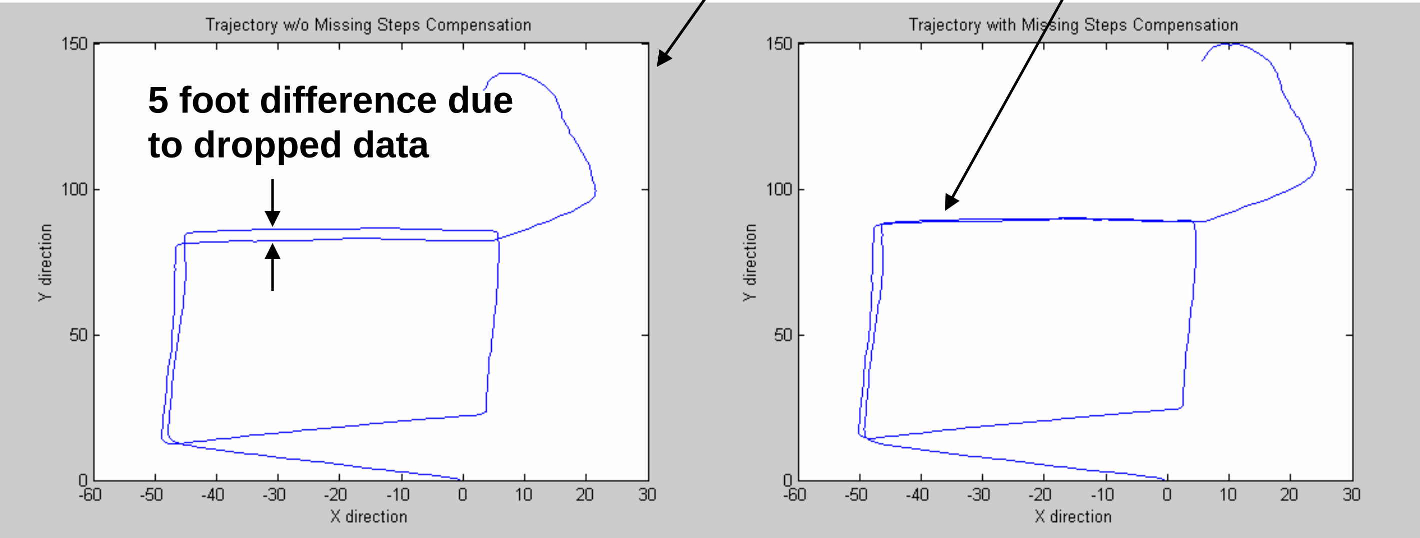


Compensated step length for missing data

Data lines may be dropped during download.

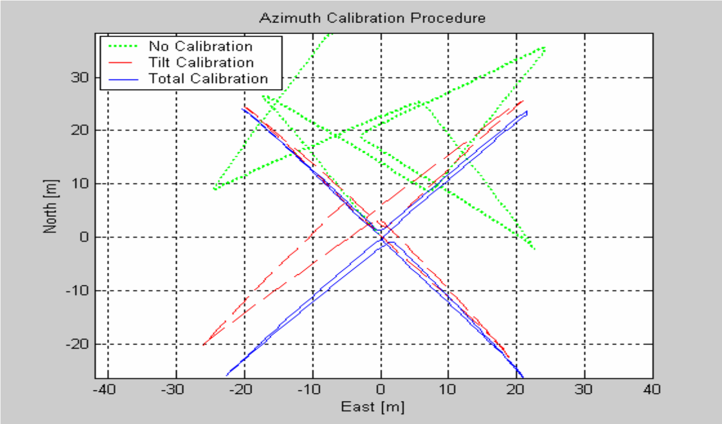
Dropped data may cause significant errors in trajectory reconstruction

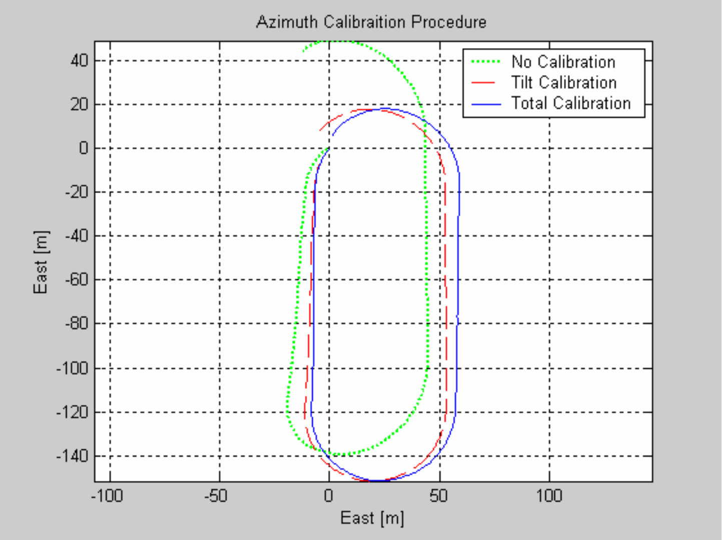
Compensated removes 5 foot error in reconstructed trajectory



5 foot difference due to dropped data

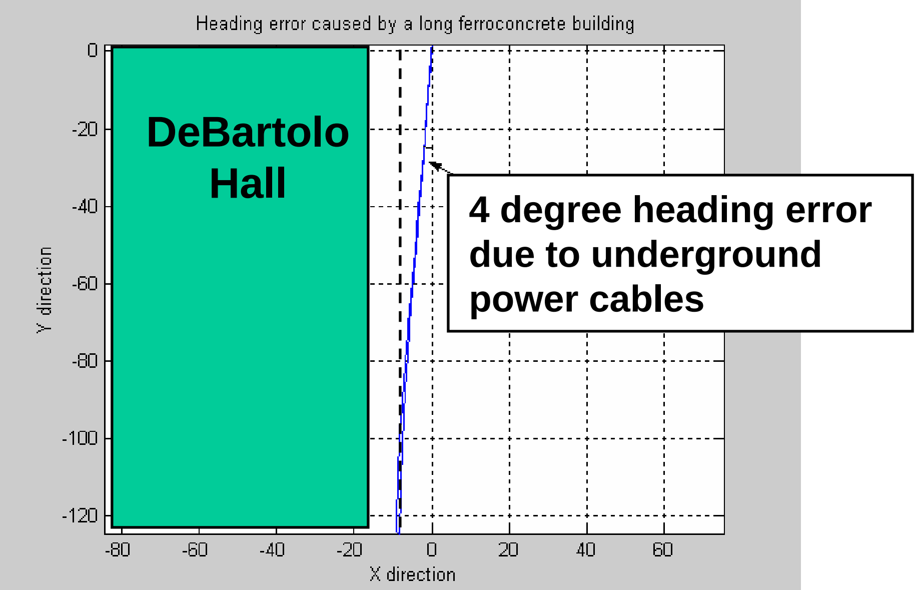
Calibration



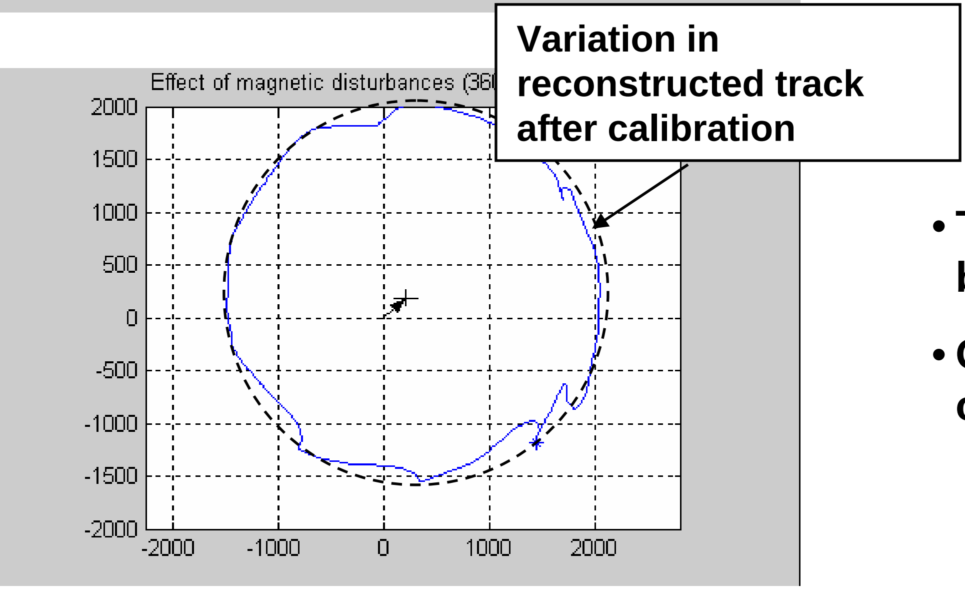


- Calibration is essential in order to obtain good performance
- NEST middleware provides a method for in situ calibration of MICA sensors.
- NEST localization services can be used to provide “absolute” position fixes that can be transmitted to TagMote.
- These absolution position fixes can then be smoothed with raw data to provide auto-calibration of the TagMote while it is in the sensor field
- Currently, we use two calibration steps.
  - Pre-calibration of Leica module
  - Post-processing to adjust trajectory to known landmarks

Error Sources



4 degree heading error due to underground power cables



Variation in reconstructed track after calibration

- Significant directional errors may occur due to local magnetic anomalies
- This may be compensated if magnetometer readings are smoothed with gyro measurements
- Other sources of error arise due to
  - Mis-calibrated sensors
  - Tag placement on individual
  - Gait variations
- These other error sources may also be compensated.
- Observed Errors over 5 minute walk on the order of 5 meters.

Outline

- Red Force Tagging
  - designed and built a sensor network with a “mobile” TagMote that logs sensor data when away from the NEST, downloads data upon returning to the NEST, and then exfiltrates data from NEST to RelayMote.
- **Wireless Sensor Networks**
  - Impact of Rayleigh fading on NEST middleware services
- Overload Management in Sensor/Actuator Networks
  - Relationship between application performance and network Quality-of-Service.
- Program Issues
  - Milestones and future plans

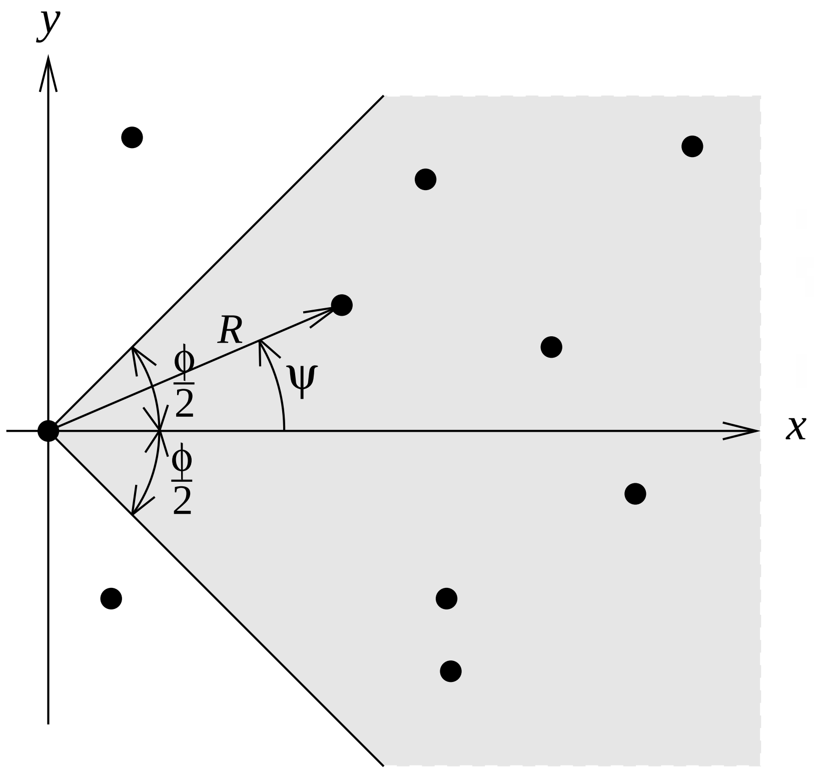
Separation Property of Rayleigh Fading Channels

For Rayleigh fading channels, packet reception probabilities can be factorized into a noise part and an interference part:

$$
\begin{aligned}
p_r &= \mathbb{P}[R_0 \geqslant \Theta(I+N)] \\
&= \exp\left(-\frac{\Theta(I+N)}{\bar{R}_0}\right) \\
&= \int_0^\infty \cdots \int_0^\infty \exp\left(-\frac{\Theta(\sum_{i=1}^k r_i + N)}{\bar{R}_0}\right) \prod_{i=1}^k p_{R_i}(r_i)\,\mathrm{d}r_1 \cdots \mathrm{d}r_k \\
&= \underbrace{\exp\left(-\frac{\Theta N}{P_0 d_0^{-\alpha}}\right)}_{p_r^N} \cdot \underbrace{\prod_{i=1}^k \frac{1}{1+\Theta \frac{P_i}{P_0}\left(\frac{d_0}{d_i}\right)^\alpha}}_{p_r^I} .
\end{aligned}
$$

So, we can focus on $\boxed{\text{noise}}$ and $\boxed{\text{interference}}$ separately.

Distances in Random Networks

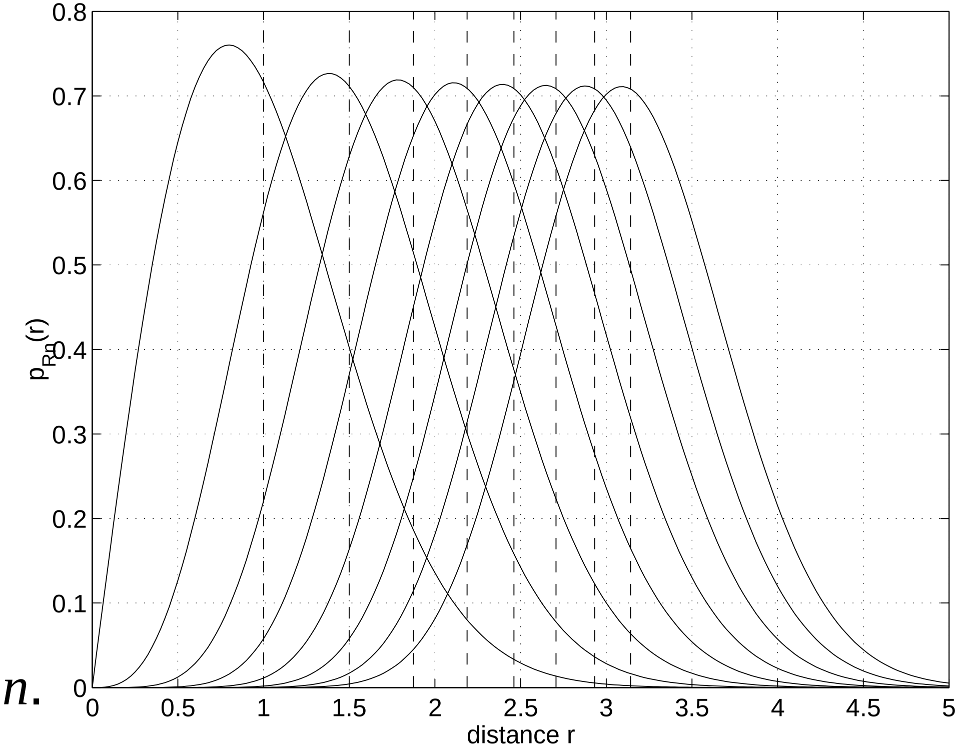


Network with uniformly randomly distributed nodes of unit density.

The *x*- axis points toward the sink.

Distance to the n-th neighbor in sector φ (generalized Rayleigh distribution):

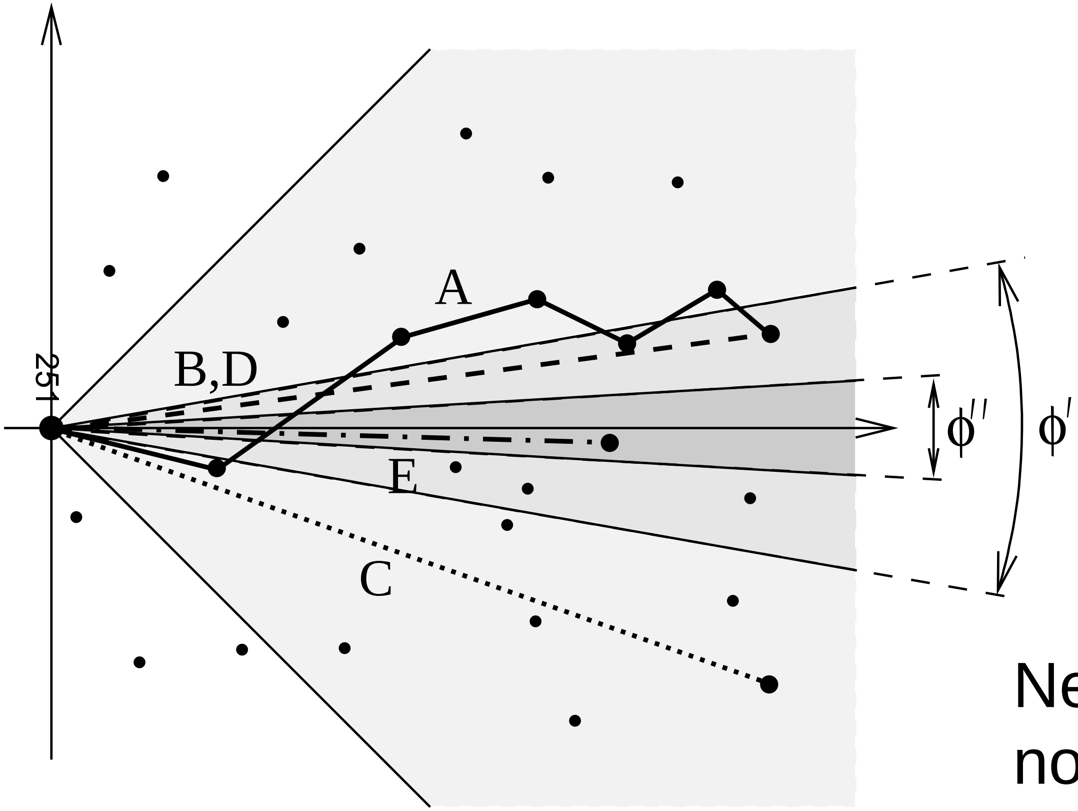
$$p_{R_n}(r) = r^{2n-1} \left(\frac{\phi}{2}\right)^n \frac{2}{(n-1)!} e^{-r^2\phi/2}$$



The variance is independent of *n*.

Routing in Rayleigh Networks

Comparison of different routing schemes:



Energy consumption:

$$E_A = n^2 \underbrace{\left(\frac{2}{\phi}\right)^{\alpha/2} \Gamma\left(1+\frac{\alpha}{2}\right)}_{\mathbb{E}[R^\alpha]}$$

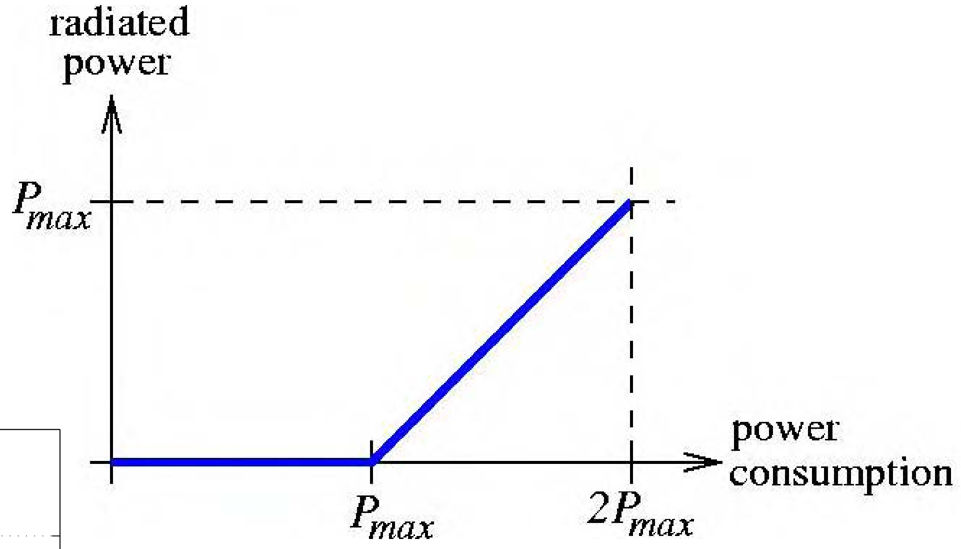
Conclusion:

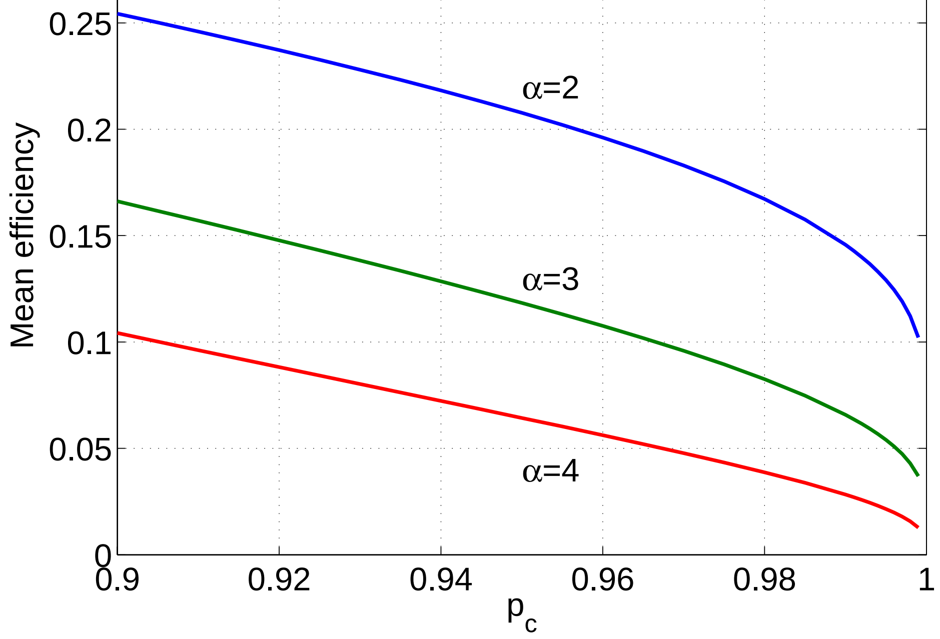
Nearest-neighbor routing is not as energy-efficient as it seems to be.

M. Haenggi, “*On Routing in Random Rayleigh Fading Networks*”, submitted to IEEE Trans. on Wireless Comm., 2003.

Connectivity and the Impact of the Power Amplifier

Generic power amplifier:
The efficiency is 50% at maximum radiated power.
Most amplifiers are worse!





PA efficiency if nodes have to be connected with high probability p_c.
Due to the long tail of the distance distribution, the efficiency is very low.

Connectivity and the Impact of the Power Amplifier

Solution: Transmit to the furthest neighbor that can be reached with high enough probability. (Again, no nearest-neighbor routing).

Distance to the furthest neighbor within distance d_{max}, (given that there is at least one node within d_{max}):

$$p_R(r) = \frac{r\phi e^{r^2\phi/2}}{e^{d_{\max}^2\phi/2} - 1}, \quad r \in [0, d_{\max}]$$

This way, the efficiency *increases* with increasing probability of being connected.

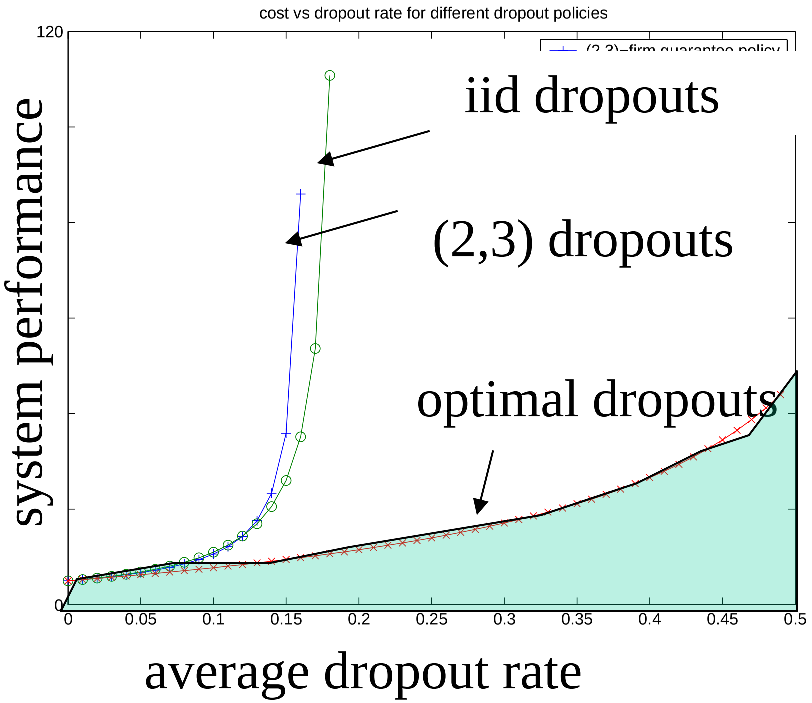
M. Haenggi, “*On the Impact of Power Amplifier Characteristics on Routing in Random Wireless Networks*”, IEEE GLOBECOM 2003.

Outline

- Red Force Tagging
  - designed and built a sensor network with a “mobile” TagMote that logs sensor data when away from the NEST, downloads data upon returning to the NEST, and then exfiltrates data from NEST to RelayMote.
- Wireless Sensor Networks
  - Impact of Rayleigh fading on NEST middleware services
- **Overload Management in Sensor/Actuator Networks**
  - Relationship between application performance and network Quality-of-Service.
- Program Issues
  - Milestones and future plans

A Fundamental Lower Bound on Achievable System Performance

- Analytical characterizations of control system performance as a function of average dropout rate.
  - independent and identically distributed dropping of packets
  - Markov dropout processes
- This characterization provides a **fundamental lower bound** on the achievable system performance
  - We can identify the “optimal” dropout process that maximizes overall control system performance
  - In certain cases we have shown that this “optimal” dropout process outperforms standard firm real-time heuristics such as skipover or *(m,k)*-guarantee rule.

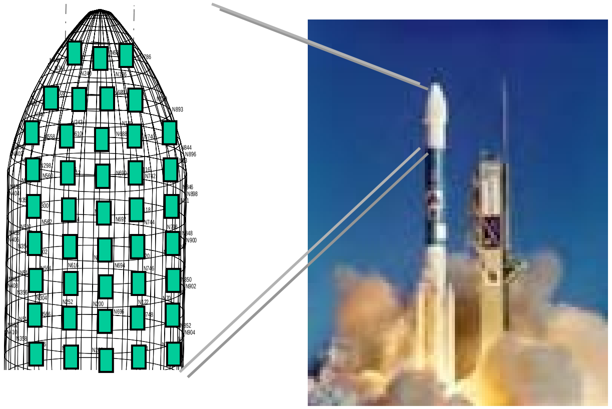


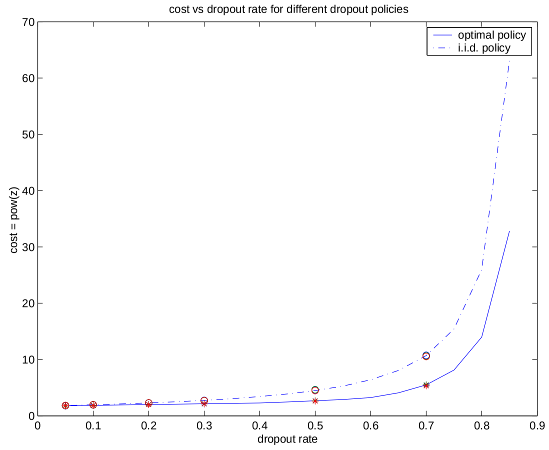
[1] Q. Ling and M.D. Lemmon (2002) “Robust Performance of Soft Real-time Networked Control Systems with Data Dropouts”, 41st IEEE Conference on Decision and Control, Dec. 2002 (also submitted for publication in IEEE Trans. On Automatic Control

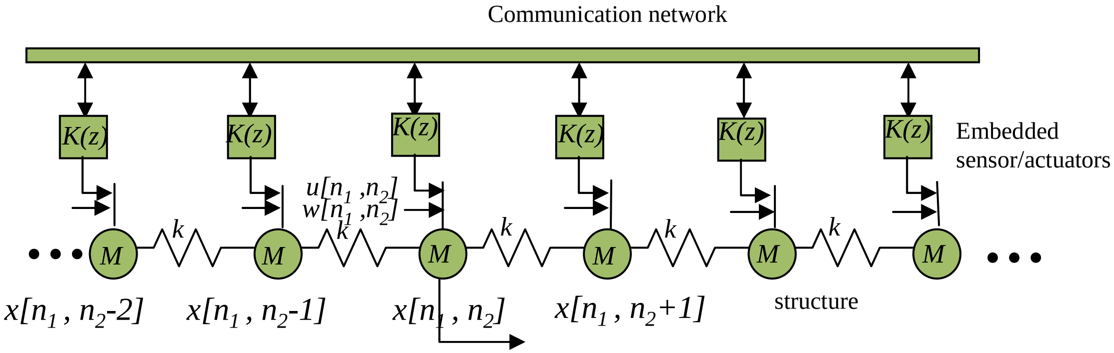
[2] Q. Ling and M.D. Lemmon (2003) “Soft real-time scheduling of networked control systems with dropouts governed by a Markov chain, American Control Conference, June 2003 (also submitted for publication in IEEE Trans. On Automatic Control

Boeing OEP Revisited

- These methods were applied to a modified version of the Boeing OEP problem
- Localized Spatially Distributed Controller Used
- Examined “overload management” scheme that purposely drops messages to assure all controllers have fair access to the network.
- Analytical way to determine “optimal” dropout sequences for distributed systems.



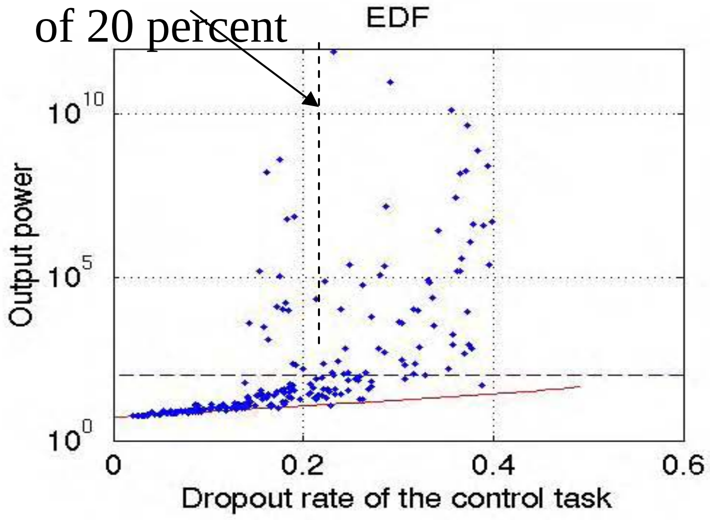




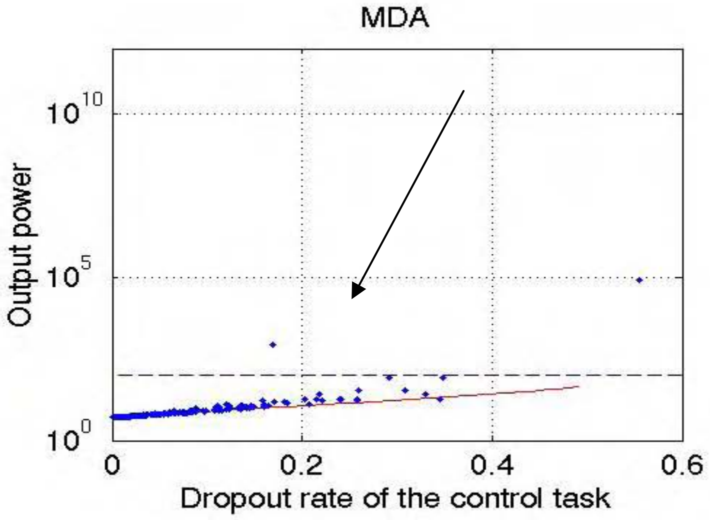
[4] M. D. Lemmon, Q. Ling and Y. Sun (2003) “Overload management in sensor-actuator networks used for spatially distributed control systems”, 1st ACM conference on Sensor Networks (SenSys03), UCLA 2003.

Novel QoS Constraint for Real-time Control

Traditional EDF scheduling results in unstable systems for dropout rates in excess of 20 percent



simple heuristic achieves QoS constraint with wider stability margins



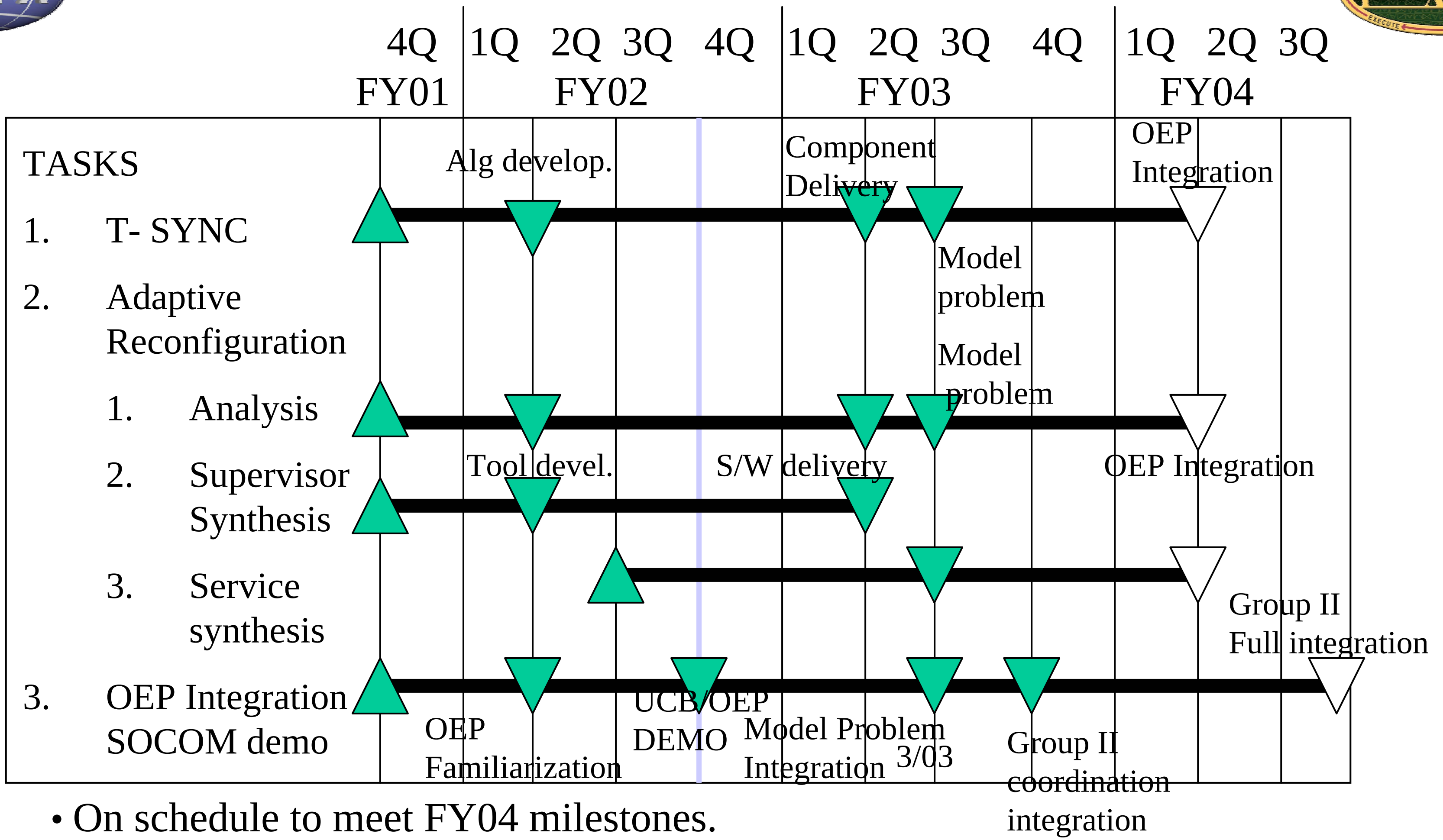
- This lower bound provides a novel of QoS constraint for firm real-time systems.
- It is possible to design several scheduling heuristics that can enforce these constraints, thereby ensuring a minimum level of overall system performance within the firm real-time environment to be found in ad hoc networks.

[3] D. Liu, Q. Ling, X. Hu, and M.D. Lemmon (2003), Firm real-time system scheduling based on a novel QoS Constraint, submitted to Real-time systems symposium 2003

Outline

- Red Force Tagging
  - designed and built a sensor network with a “mobile” TagMote that logs sensor data when away from the NEST, downloads data upon returning to the NEST, and then exfiltrates data from NEST to RelayMote.
- Rayleigh Networks
  - Impact of Rayleigh fading on NEST middleware services
- Overload Management in Sensor/Actuator Networks
  - Relationship between application performance and network Quality-of-Service.
- **Program Specifics**
  - **Milestones, future plans, issues**

Project Schedule and Milestones



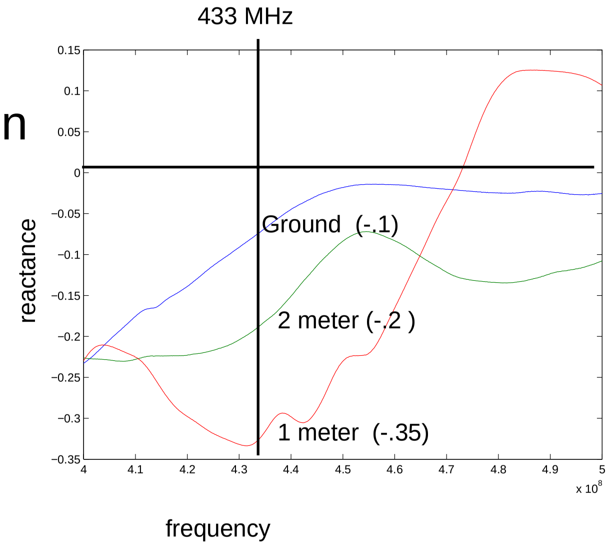
- On schedule to meet FY04 milestones.
- SOCOM demo.

Project Plans

- Plan over next 6 months
  - Red Force Tagging
    - Characterize size/cost for TagMote using near-term Technologies
    - Improve download from Tag-Nest using Semantic Compression
    - Develop and Test Auto-calibration schemes.
    - Develop strategy for automatically downloading data to net.
  - NEST Services
    - Improve download time for current services
    - Integrate Localization into service
- Success Goals
  - TagMote to NEST download = (1 min walk to 15 seconds of download time)
  - NEST to RelayMote (exfiltration) = 256 bytes/second (sustained)

Program Issues

- **Tech Transition/Transfer**
  - SoCom Demonstration at MacDill AFB
- Issues:
  - Ground effects in Mica2 radios. Impedance mismatch to antenna appears to significantly reduce Chipcon radio range.



NEST SOCOM

RED FORCE TAGGING

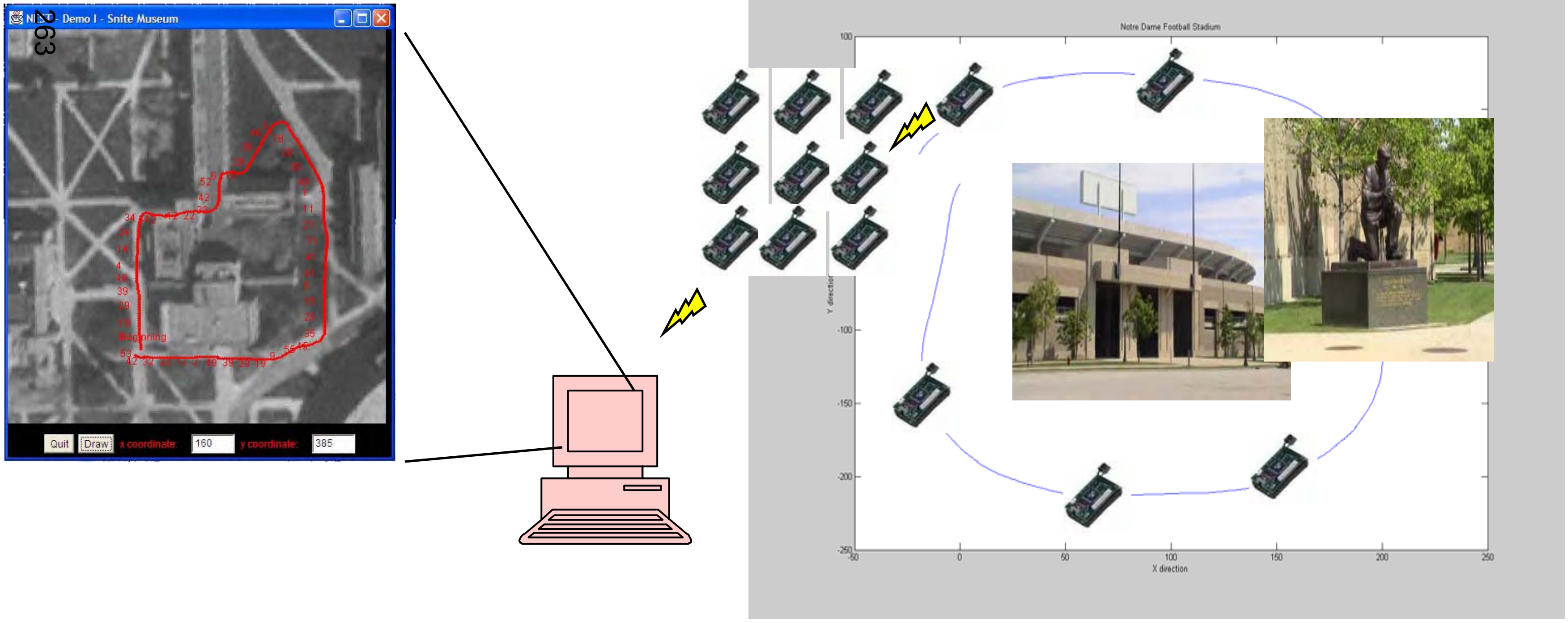
Technical Issues

July 14, 2003

P.J. Antsaklis, *M.D. Lemmon, M. Haenggi*
University of Notre Dame

Problem Overview – Red Force Tagging

- The objective is to reconstruct the space-time trajectory of a “tagged” individual as they move through GPS denied regions.
- The approach involves “tagging” the individual with a special mote (TagMote) that records subject’s acceleration and direction.
- The TagMote then downloads this data to the NEST and the NEST forwards the data to a RelayMote where trajectory reconstruction is done.

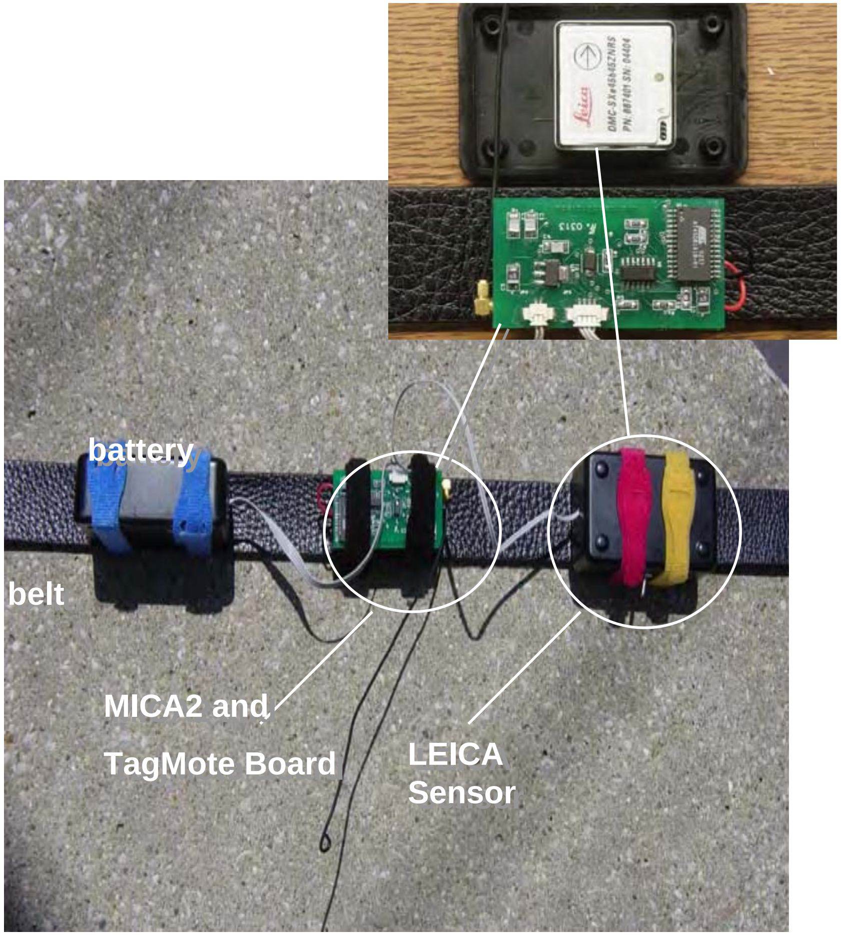


Detailed Scenario Description

- **Setup:** The network grid is physically deployed, initialized and synchronized so it can receive data from a mobile mote (TagMote) that joins the network, and data that have been downloaded at any mote can be routed to the designated special Relay mote.
- **Initialization:** During initialization, the special device (TagMote) acquires its position and time to be used as reference from the network. For the demo, initial position and time are assumed to be known.
- **Operation:** A subject carrying the special TagMote walks out of range of the network for some time interval, walks in a normal fashion tracing some path and returns within range of the network at some later time.
- **Data transmission to the network:** The data from the TagMote are downloaded wirelessly to a network mote using NEST middleware services.
- **Data routed to the relay mote within the network:** NEST services are used.
- **Data processing, display and evaluation:** Data transmitted to the Relay mote are processed and displayed to show the trajectory traced by the subject while out of range of sensors.

Technical Approach – Red Force Tagging

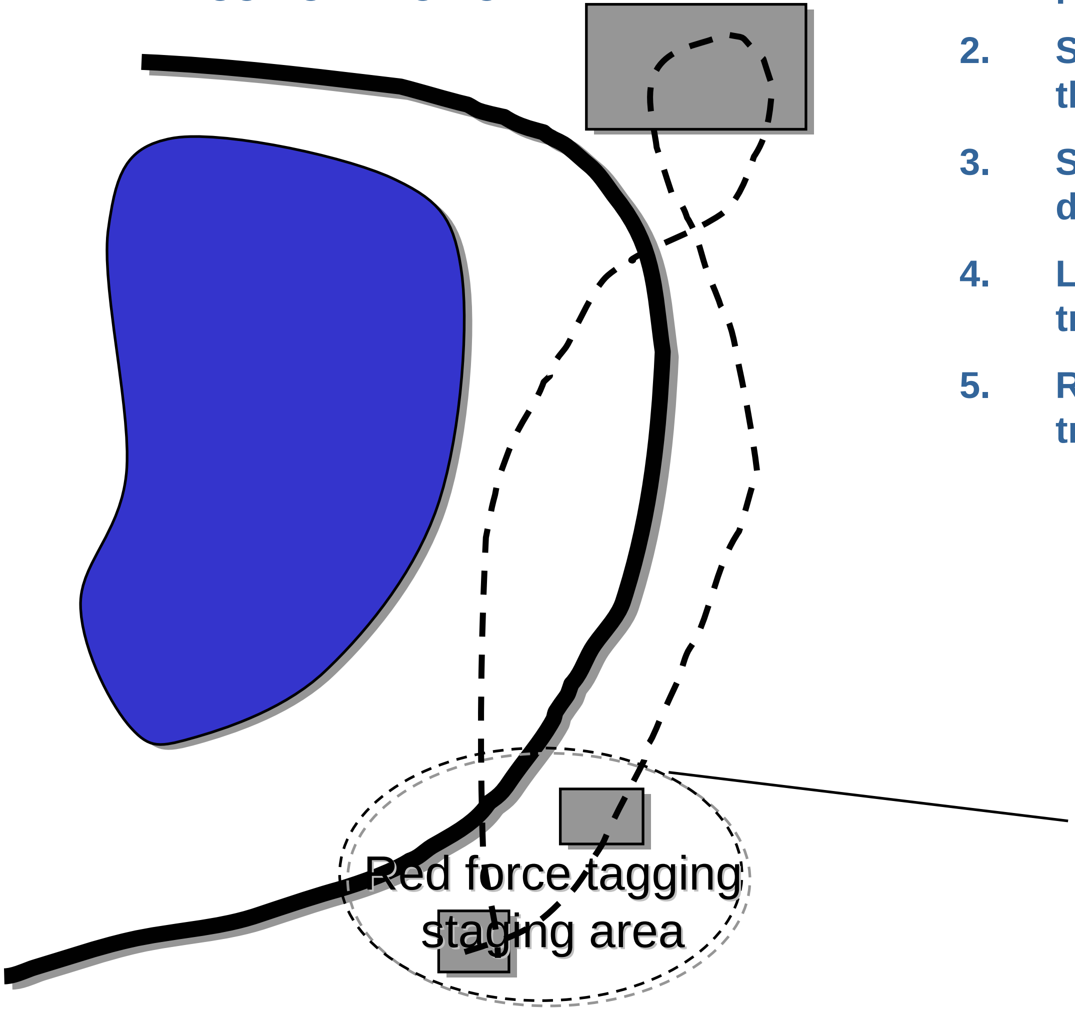
- Design and Build a special “TagMote”
  - MICA2 for communication with the NEST
  - Leica Sensor
    3D-accelerometer
    magnetic compass
  - Specially designed TagMote board with 4 MByte Flash Memory
  - currently attached to a belt that is worn by the subject

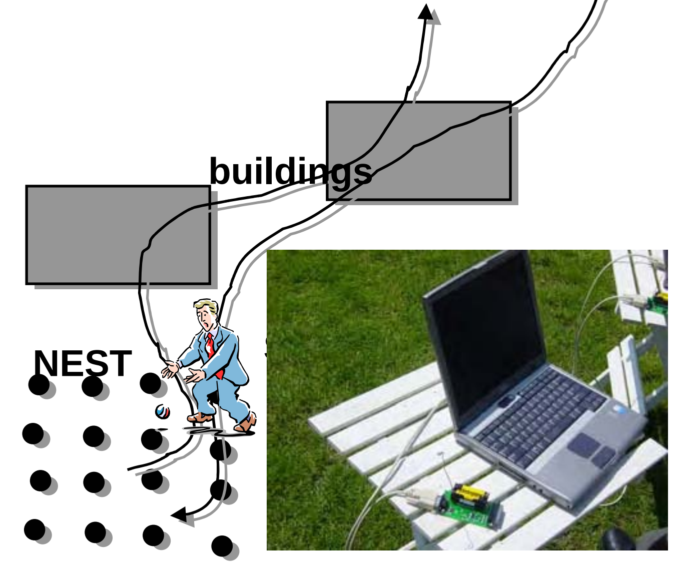


Red Force Tagging Physical Layout

- Physical layout of RedForce Tagging Staging Area

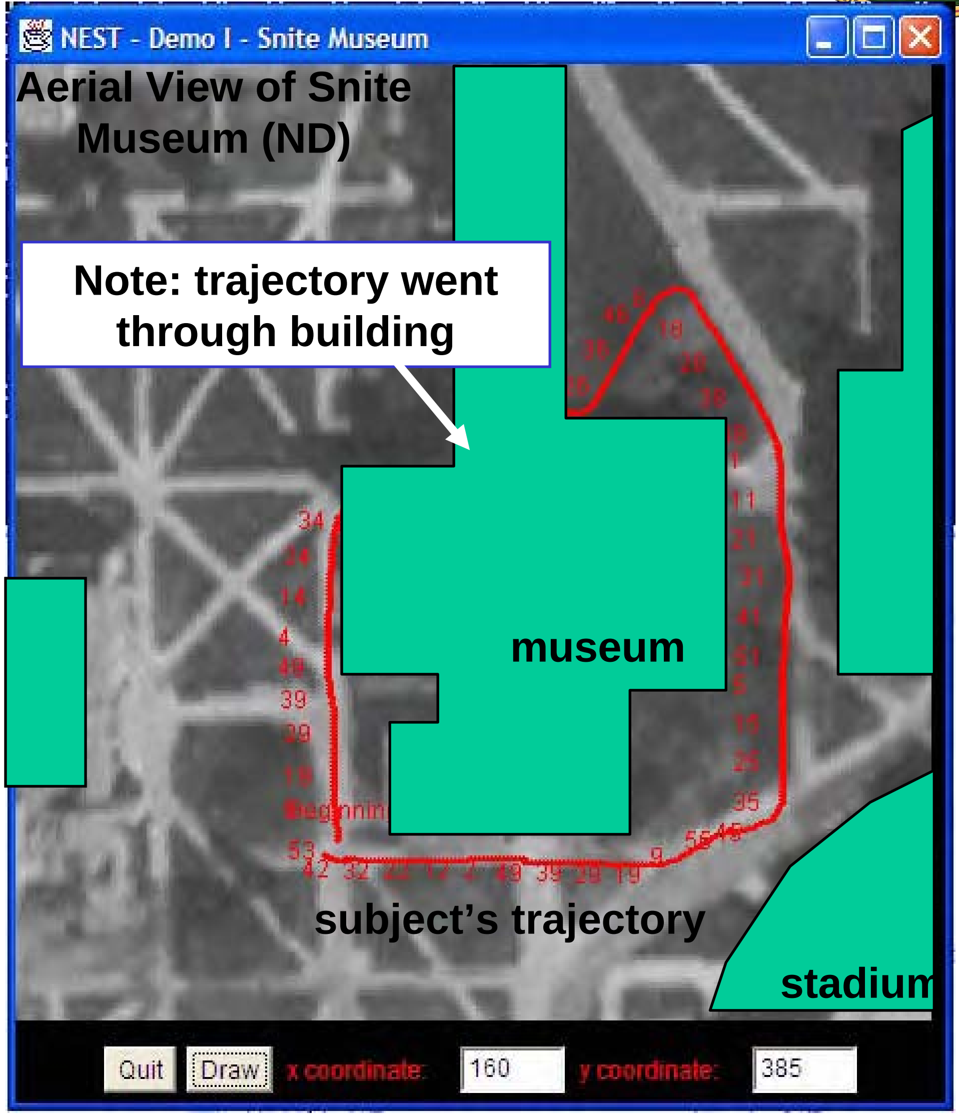
1. Subject attaches TagMote Belt and moves to initial starting position
2. Subject takes a 3-5 minute walk through the test course
3. Subject returns and data downloaded to NestMotes
4. Listener displays reconstructed trajectory
5. Relay displays reconstructed trajectory





Trajectory GUI

- Data captured by NestConsole GUI is written to a file
- This data is processed by a Java program to generate the subject's trajectory
- The trajectory data is then plotted against an aerial view of the area
- This particular plot shows trajectory reconstruction in a GPS denied area (within bldg).



Aerial View of Snite Museum (ND)

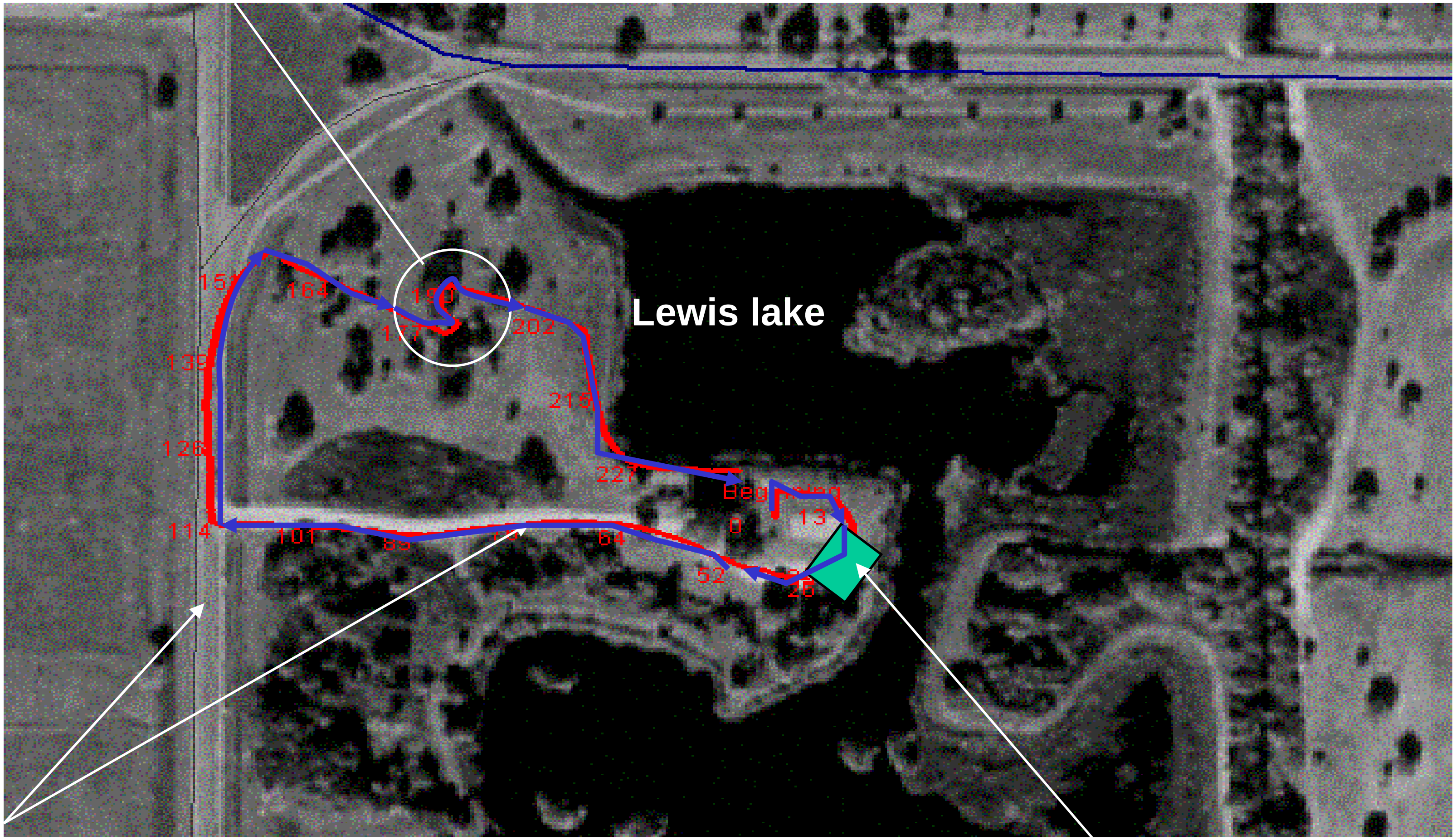
Note: trajectory went through building

x coordinate: 160

y coordinate: 385

MacDill Demonstration

maneuver to avoid water



road

structure

Detailed Scenario Description –Software

- **TagMote Software implements:**
  - Compensation for adverse sensor orientation, calibration, filtering
  - Collection of data from sensors and storage on a non-volatile memory
  - Retrieval of data from memory, processing and transmission
  - Detection of a mote network and data download
  - Pre-processing of Raw Sensor Data before data download
- **NEST network services**
  - synchronization: establishes global time reference
  - backbone generation: establishes min-hop route to RelayMote
  - streaming service: streams large blocks of data from NestMote to RelayMote
  - download service: streams large blocks of data from TagMote to NestMotes
- **Trajectory Post-Processing**
  - Java programs on PC connected to RelayMote
  - Reassembly of data packet
  - Compensation for missing data
  - step-length estimation, direction of travel estimation
  - Reconstruction of complete trajectory
  - Display of Trajectory over area map.

Technical Issues

- **Localization:**

  Not implemented at this point.
  Could improve accuracy and simplicity.

- **Radio range:**

  Is a big issue, since effective range varies substantially depending on the environment and elevation (see impedance measurements)
  varying link quality increases download time substantially

- **Sensors:**

  Magnetometer requires calibration

Relation to other NEST Projects

- **Common with other NEST Projects**
  - –Uses NEST services for network synchronization, localization, routing.
  - –The mote network could be the sensor network used for the other NEST/SOCOM projects.

- **Different from Other NEST Projects**
  - –TagMote. Uses a special mobile sensor mote that collects data which are automatically downloaded to the mote network when in proximity
  - –The mote network is used as a communication network not as a sensor network, so different density requirements.
  - –Routing of different class of data. Large amounts of data from many sources, typically the motes at the edges of the network, to single destination, the Relay mote.

Real-time Configuration of Networked Embedded Systems

P.J. Antsaklis, Martin Haenggi, M.D. Lemmon
Dept. of Electrical Engineering
University of Notre Dame

DARPA/IXO NEST PI MEETING
December 2003, Santa Fe NM

Real-time Reconfiguration of Networked Embedded Systems

University of Notre Dame

Contract No. F30602-01-2-0526
AO No: L546/00

Period: 5/2001 – 9/2004
Panos J. Antsaklis (PI)
Department of Electrical Engineering
PI Phone No: 574-631-5792, Fax 574-631-4393, antsaklis.1@nd.edu

DARPA Program Manager: Vijay Raghavan
Agent name & organization: Juan Carbonell, AFRL/Rome

Subcontractors and Collaborators

Senior Personnel

Panos J. Antsaklis, Michael D. Lemmon, Martin Haenggi

Electrical Engineering, University of Notre Dame

Graduate Students

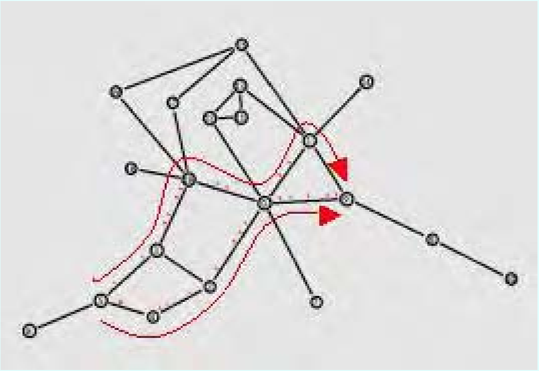
Luis Montestruque, Brett McMickell, Yashan Sun, Lei Fang, Hui Fang, Marian Iordache, Ioannis Koutroulis, Min Xie, Nicholas Kottenstette

Univ. of Notre Dame

Real-time Reconfiguration of Networked Embedded Systems

Problem and Challenge

Real-time reconfiguration of middleware services in networks of embedded systems



Real-time+ Distributed Algorithms + Feedback Loops+Probabilistic Analysis+Formal Verification

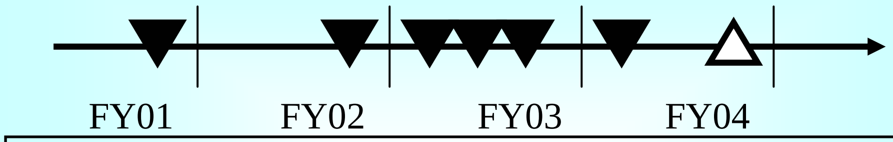
New Ideas

- Model-based clock synchronization and security for heterogeneous networks of embedded systems
- Adaptive reconfiguration for robust scalable middleware services in response to abrupt network changes
- Real-time network management through probabilistic analysis and formal verification.
- Exploiting feedback in distributed algorithms

Impact

- Middleware for ad hoc networks of embedded systems.
- Real-time reconfiguration of streaming connections in ad hoc wireless networks.
- Red Force Tagging

Schedule



3QFY01: Project Start
3QFY02: Berkeley OEP Demo
1QFY03: Demo of Model Problem (robot formation control)
2QFY03: Delivery of TinyOS Service Components
3QFY03: Integration with Group I (Red ForceTagging)
3QFY04: Challenge Problem Enhancements
4QFY04: Technical Transfer to Military

Problem Description and Objectives

Adaptive reconfiguration in fine grained networks of embedded systems that interact tightly with the physical world so to maintain Quality of Service (QoS.)

Project Tasks:

- Clock-synchronization:
  Develop clock-sync algorithms for ad hoc networks.

- Adaptive Reconfiguration
  Fast reconfiguration of NEST services
  in the presence of abrupt network changes

- Red Force Tagging for SOCOM

Project Status

PROGRESS SINCE LAST MTG.

- Red Force Tagging
  - Field Experiment at MacDill AFB
  - Download service robustness and throughput enhancements
  - TagMote hardware enhancements
- NEST Middleware Services
  - Exfiltration service robustness and throughput enhancements
- Overload Management in Sensor/Actuator Networks
  - Relationship between application performance and network Quality-of-Service
- Large Networks of Embedded Sensors
  - Power consumption and throughput in regular and random networks

Goals and Success Criteria

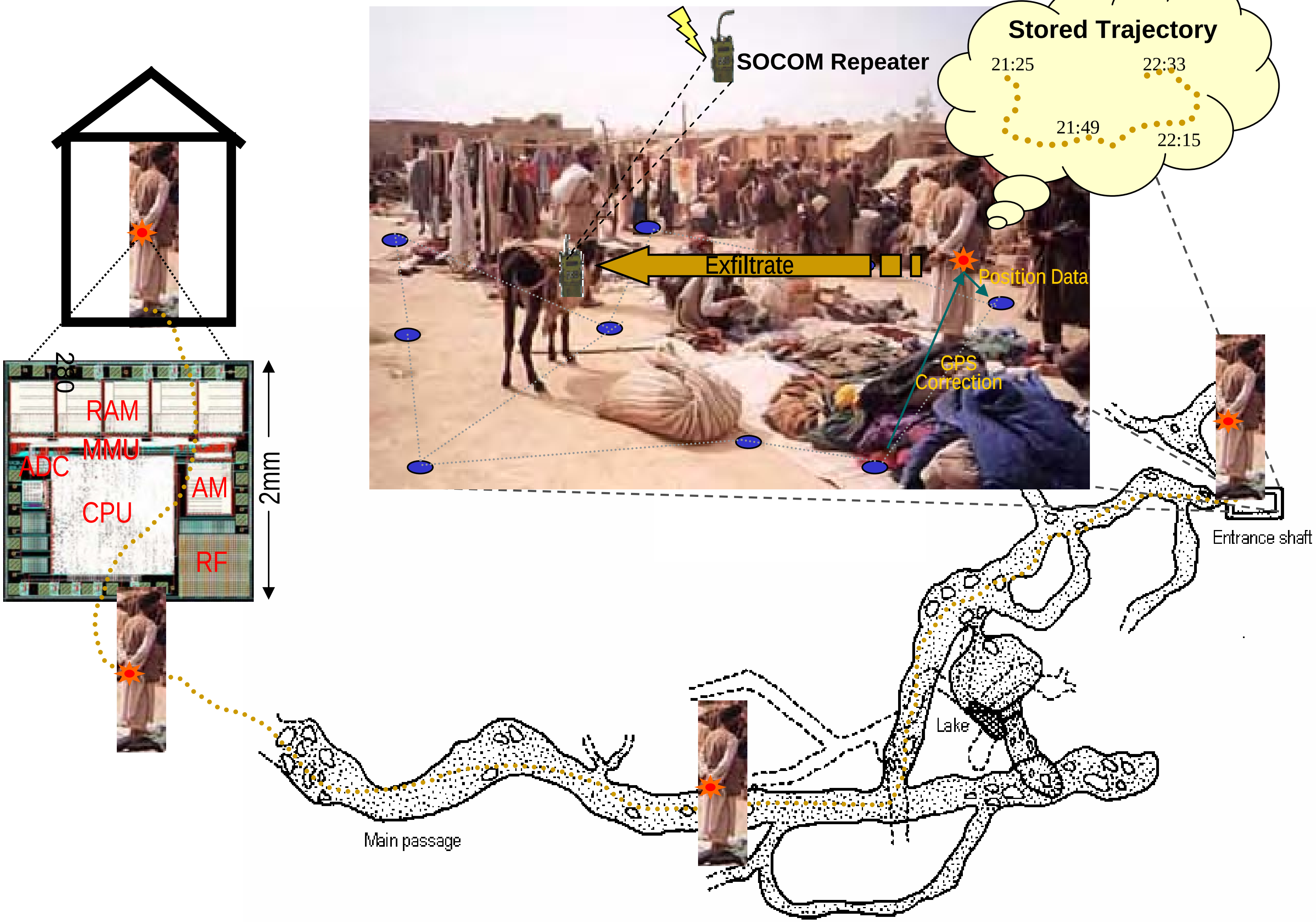
- Red Force Tagging
  - Hardware enhancement-design and implementation of microprocessor board
  - Download service robustness and throughput enhancements-test acknowledgement policies, avoid isolated network motes
- NEST Middleware Services
  - Exfiltration service robustness and throughput enhancements-Provide a highly connected routing structure
- Overload Management in Sensor/Actuator Networks
  - Analytical way to determine “optimal” dropout sequences for distributed systems
- Large Networks of Embedded Sensors
  - Identify efficient communication policies (power and throughput) in random networks

Outline

- Red Force Tagging
  - Project Description
  - Enhancements
  - Field Experiment – later today
- Overload Management in Sensor/Actuator Networks
- Large Networks of Embedded Sensors
- Program Issues

"Red Force Tagging"

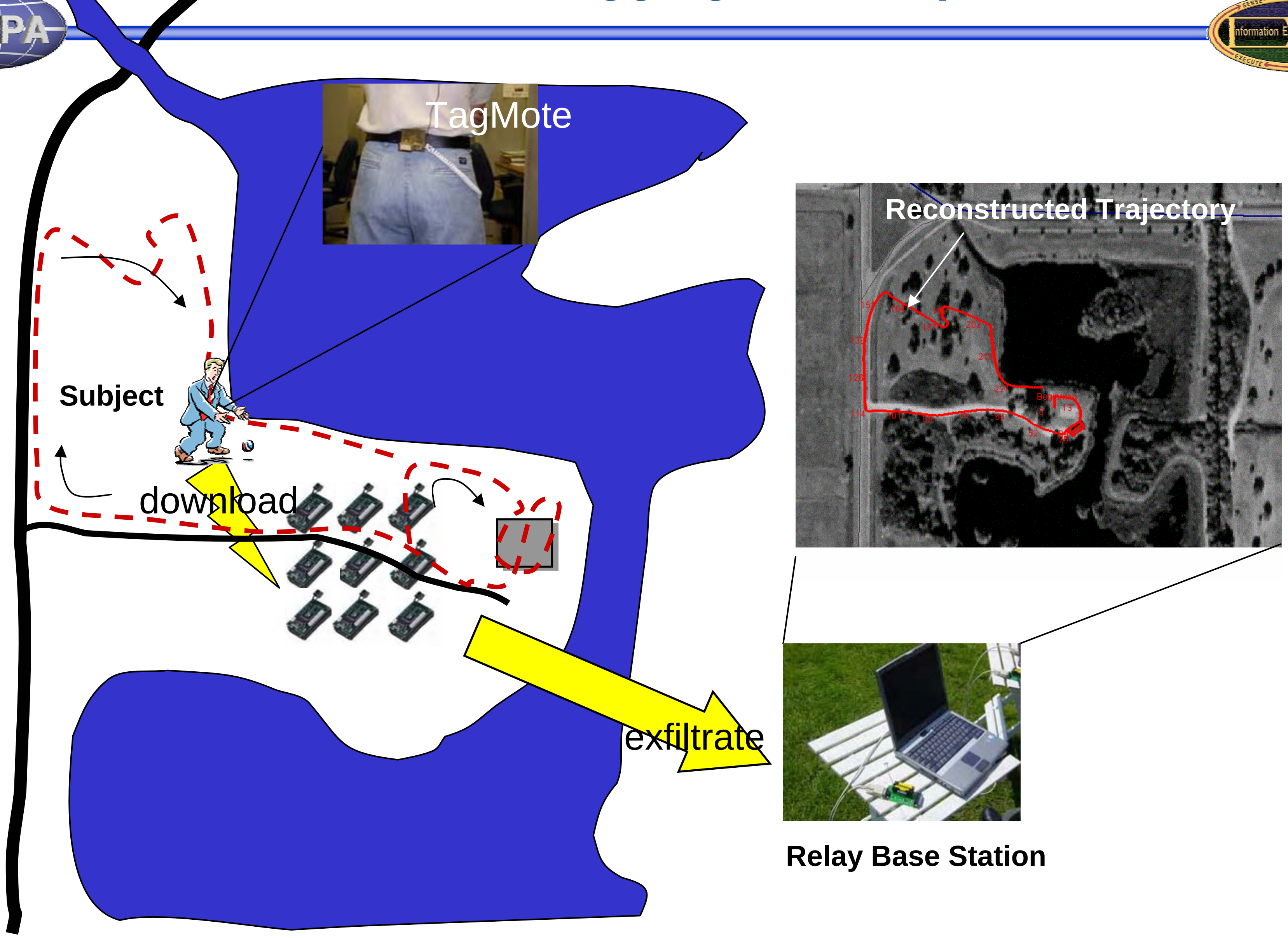
Track subjects to gather pre-conflict intelligence



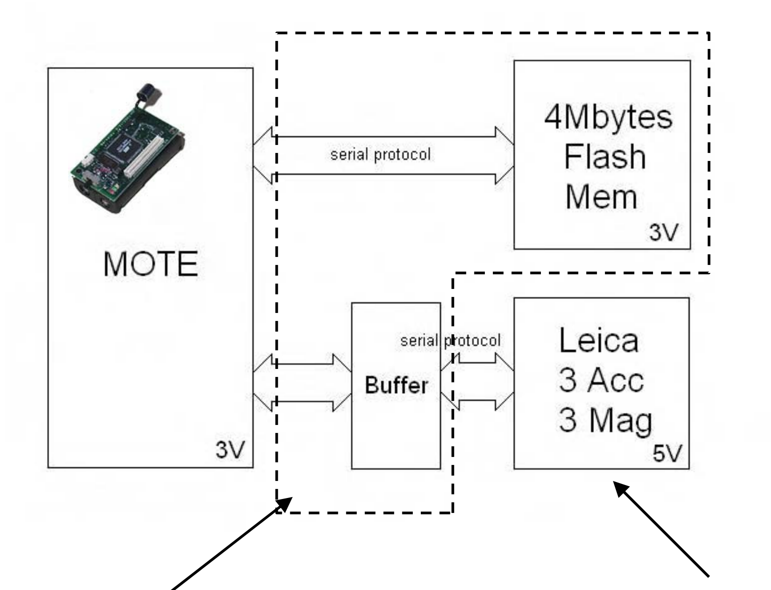
Benefits

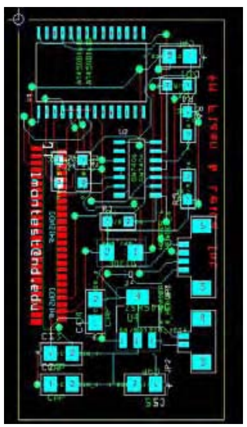
- Discovering Red Force in GPS-denied areas
- Does not rely on pre-deployed wired network
- Can be used by Blue Force
- Multi-sensor data collecting platform
  - Extensible to other sensor types : biochem, radiation

Red Force Tagging Field Experiments



TagMote Hardware Overview



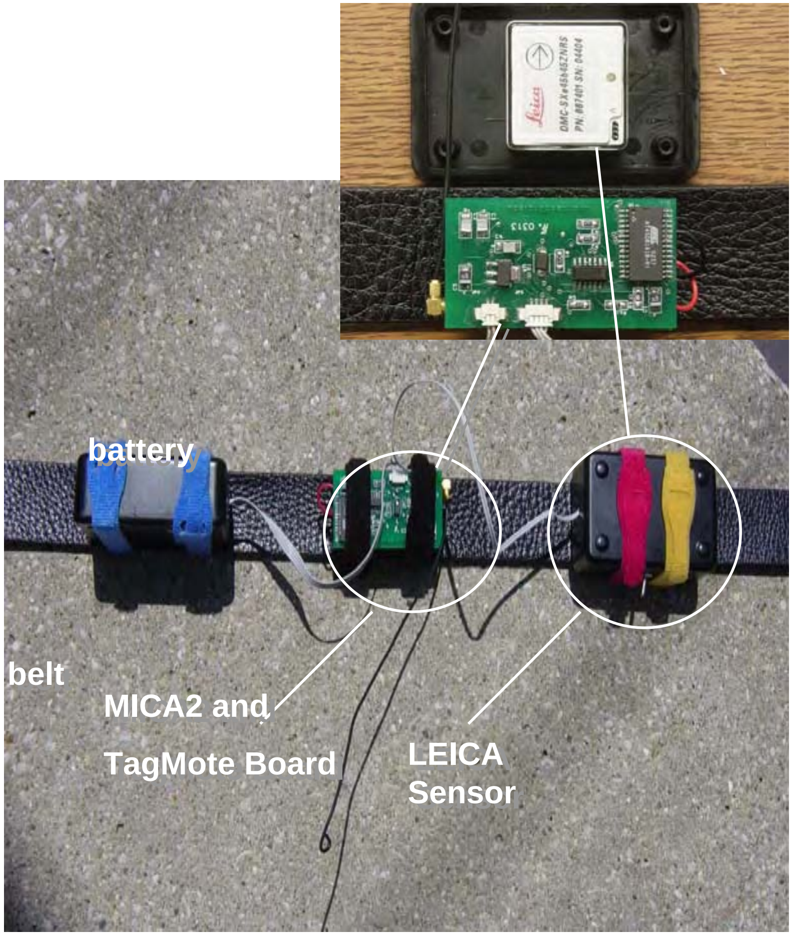


TagMote Board

Leica Module

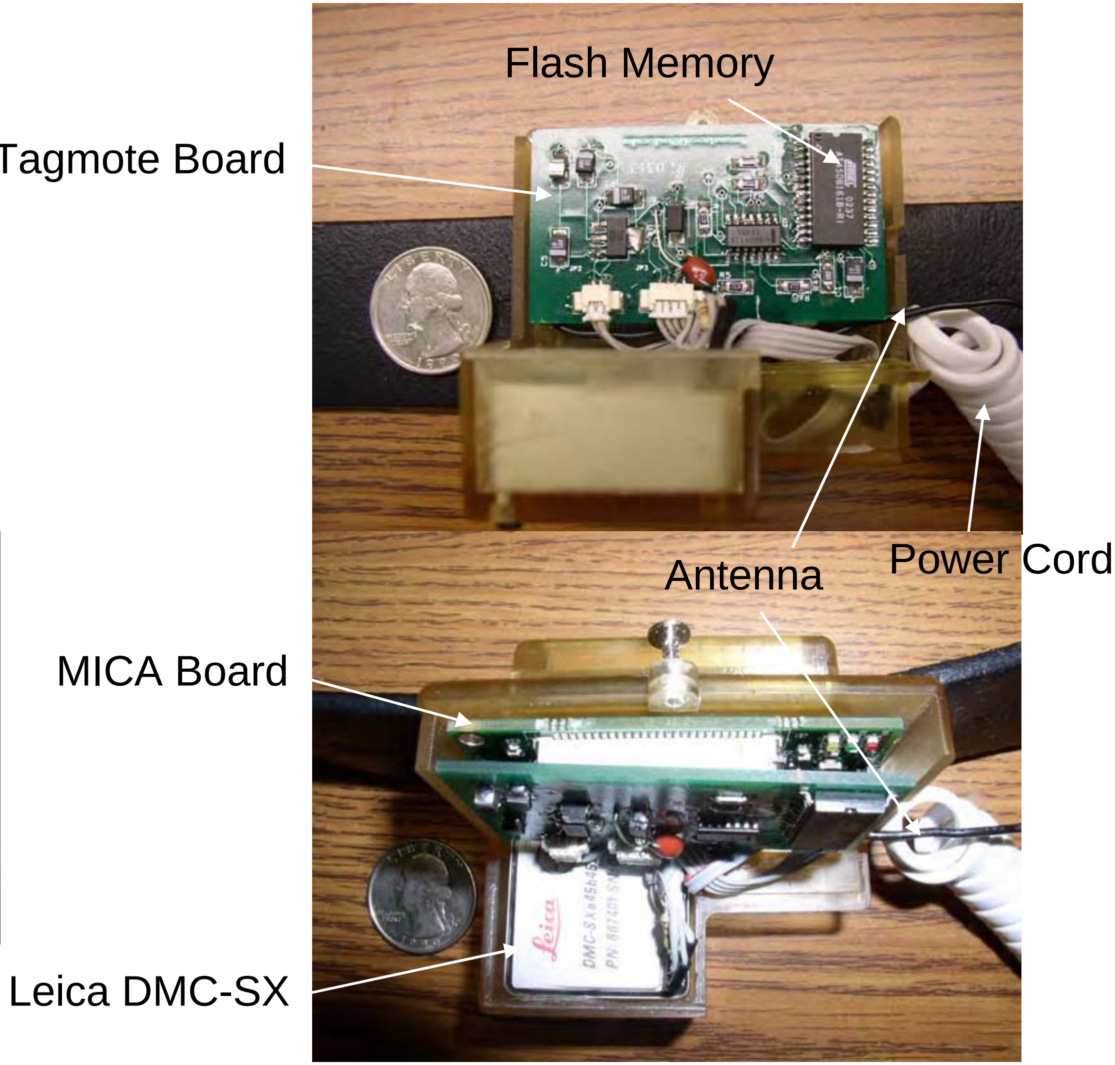
How it is done?
Technical Approach – TagMote

- Designed and Built a Special “TagMote”
  - TagMote board
    4 MByte Flash Memory
  - Leica Sensor
    3D-accelerometer
    magnetic compass
  - MICA2 for communication
  - Attached to a belt



TagMote: Current Design

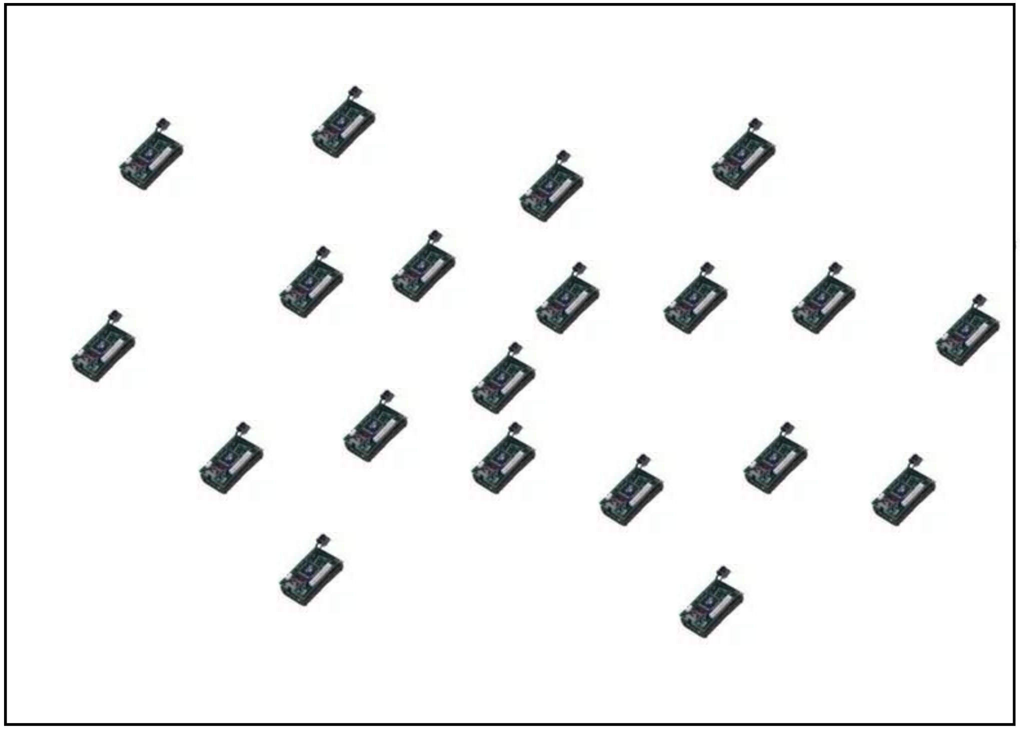
- Compact: 65 ×58×24 mm
- Lightweight:100 grams



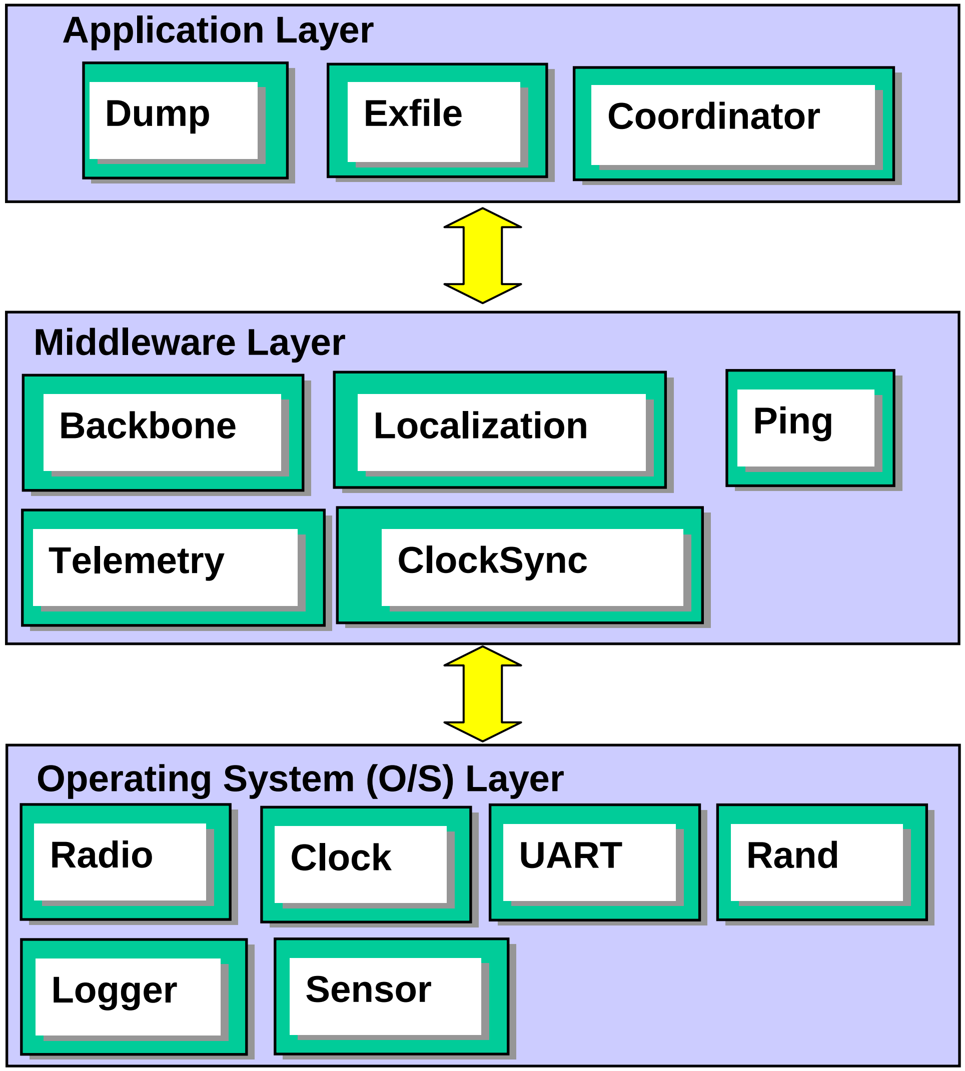
Technical Approach – NEST Middleware

- NEST Middleware Challenges
  - Download large amounts of data from TagMote to NEST
  - Exfiltrate large amounts of NEST data to SOCOM relay
  - Reliable and Timely Data Delivery
- Solutions
  - Multi-hop Streaming service (link reservation)
  - Explicit packet acknowledgement
  - Route reconfiguration (based on link level timeouts)

Clear Basic Service Cell



Red Force Tagging Middleware Architecture



Application Layer

Dump

Exfile

Coordinator

Middleware Layer

Backbone

Localization

Ping

Telemetry

ClockSync

Operating System (O/S) Layer

Radio

Clock

UART

Rand

Logger

Sensor

TagMote Software Tasks

- **Tagmote software implements:**
- Compensation for adverse sensor orientation, calibration, filtering;
- Collection of data from sensors and storage on a non-volatile memory;
- Retrieval of data from memory, processing and transmission;
- Detection of NEST and data download;
- Pre-processing of raw sensor data before data download.

- **NEST network services provide**
- Synchronization: establishes global time reference;
- Backbone generation: establishes min-hop route to RelayMote;
- Streaming service: streams large blocks of data from NestMote to RelayMote;
- Download service: streams large blocks of data from TagMote to NestMotes.

- **Trajectory Post-Processing implements**
- Java programs on PC connected to RelayMote;
- Reassembly of data packet;
- Compensation for missing data;
- Reconstruction of complete trajectory;
- Display of trajectory over area map.

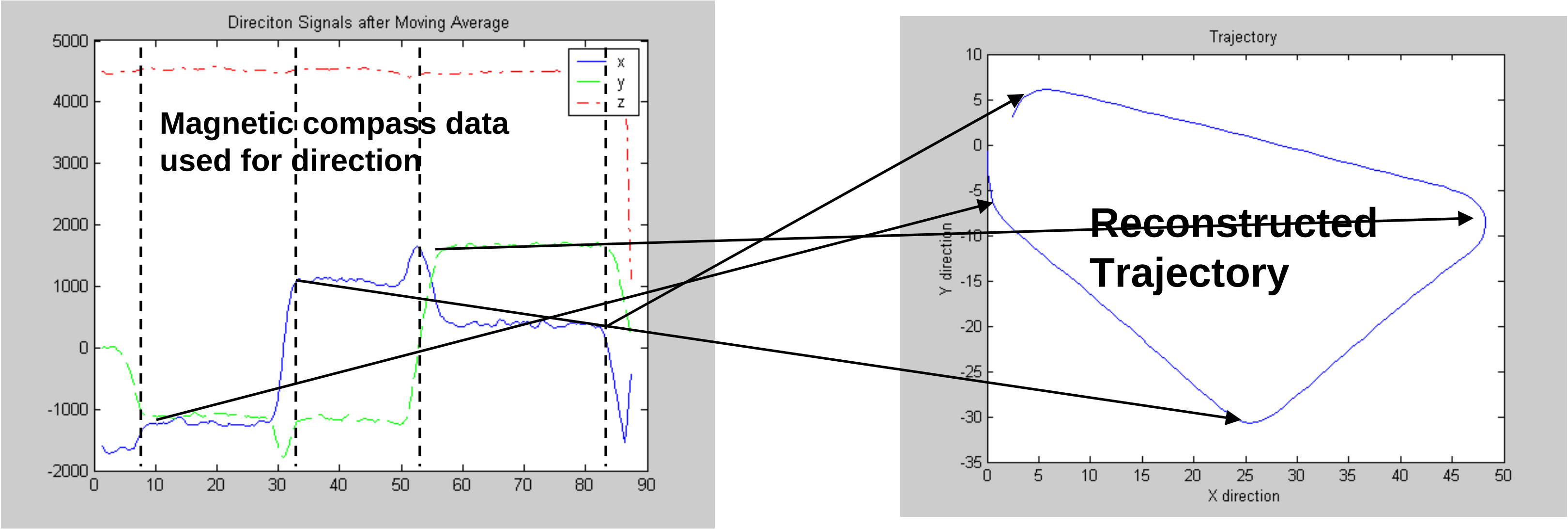
Technical Approach – Algorithms
Dead Reckoning

Data Generated by TagMote

- **Step length estimation**

$$K \times \sqrt[4]{a_{z\max} - a_{z\min}}$$

- Data from accelerometer and magnetic compass
- Trajectory reconstructed from step count, step length, and direction
- Part of processing done on TagMote
  Bulk of trajectory reconstruction done at RelayMote



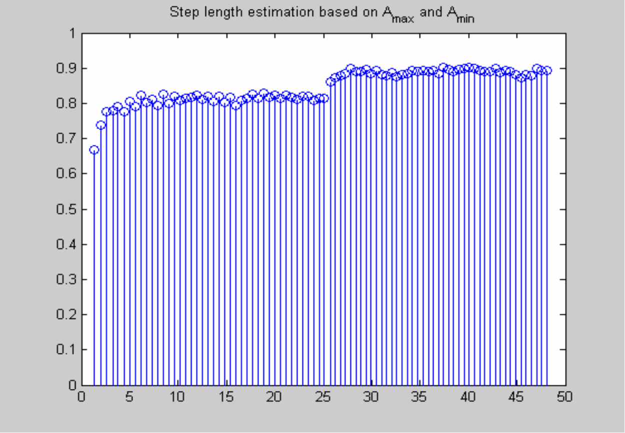
Magnetic compass data used for direction

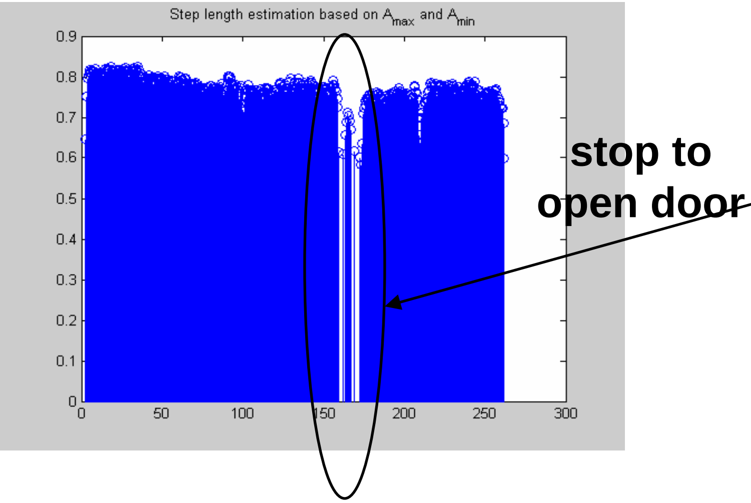
Reconstructed Trajectory

Trajectory Reconstructed at RelayMote

Gait Analysis

Change in speed & stride length

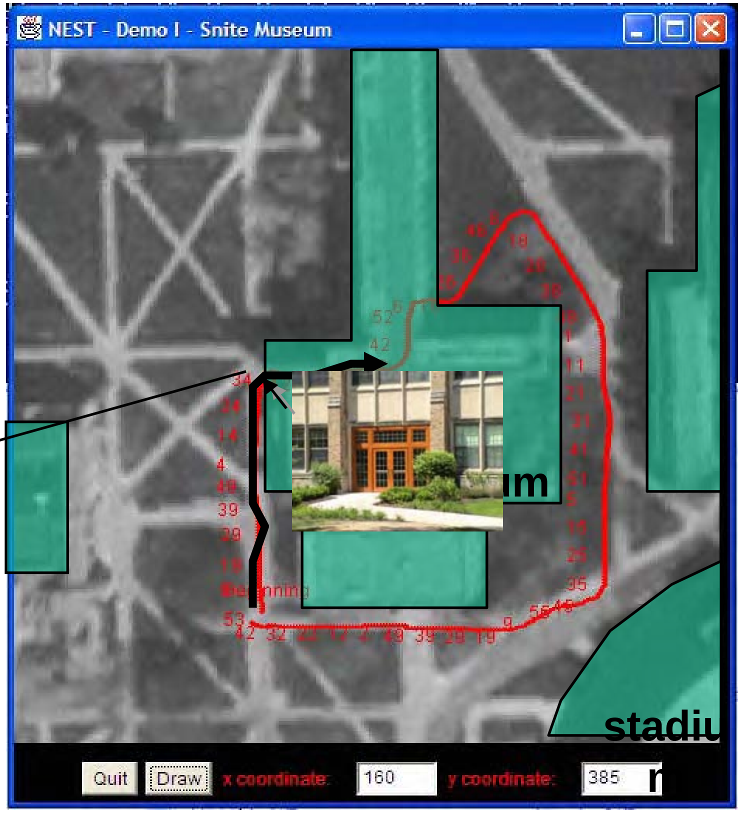




stop to open door

Acceleration and step length histories can be used to analyze subject’s gait

- gait changes walking/running
- stopping to open doors

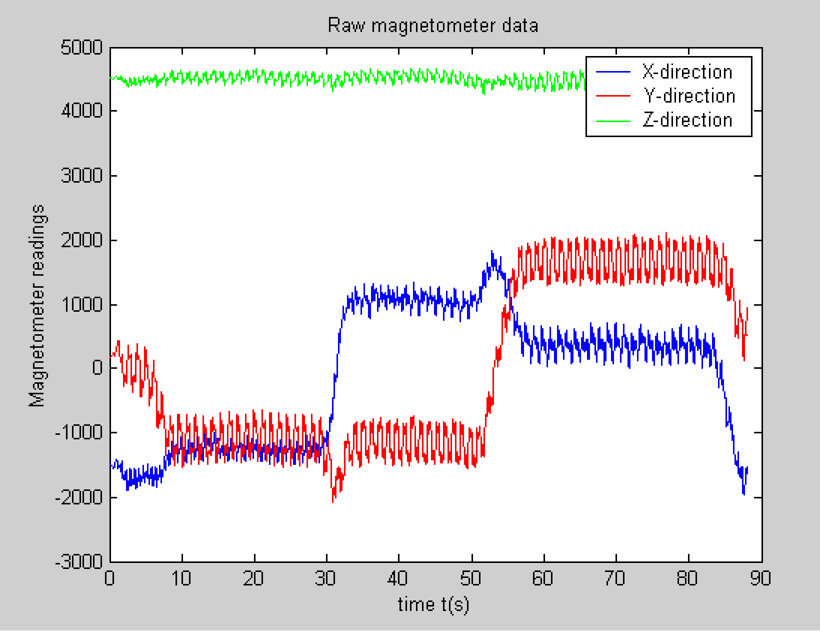

NEST - Demo I - Snite Museum

Quit Draw x coordinate: 160 y coordinate: 385

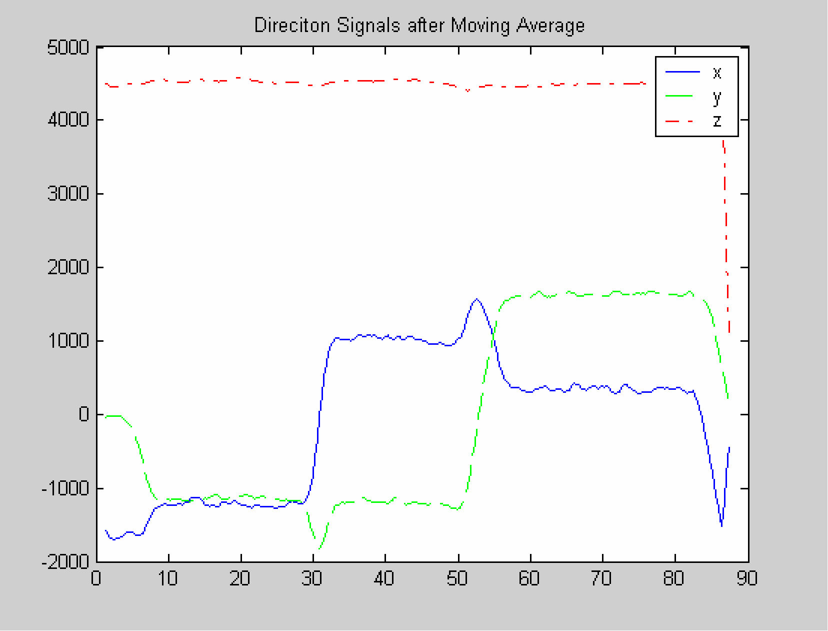
Heading Estimation

- Moving-average

Raw Compass Data

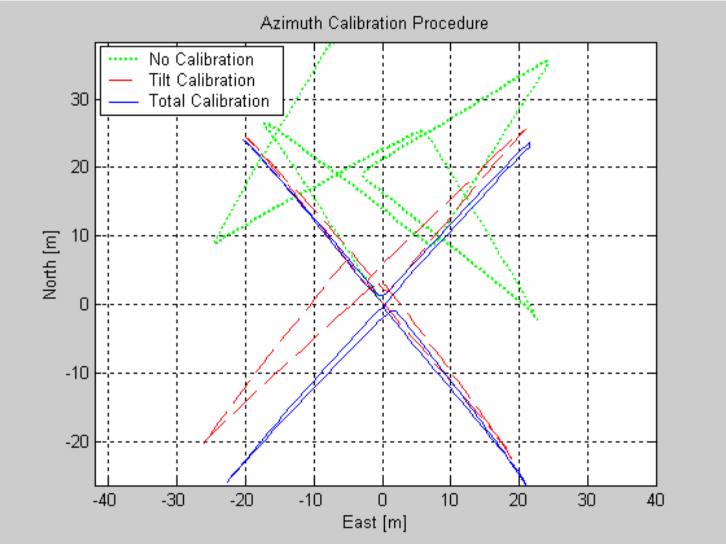


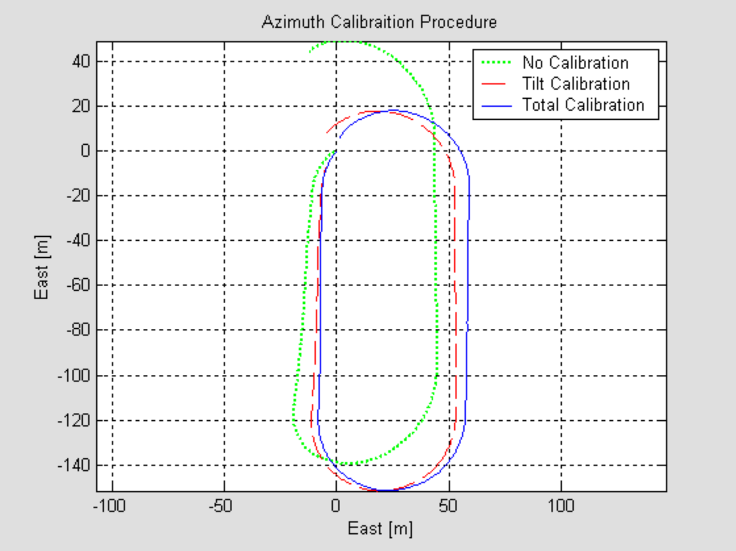
After Moving-average



- Dynamic calibration: Tilt compensation using 3-D accelerometer data
- Static calibration: Magnetic field distortion (e.g., due to human body)

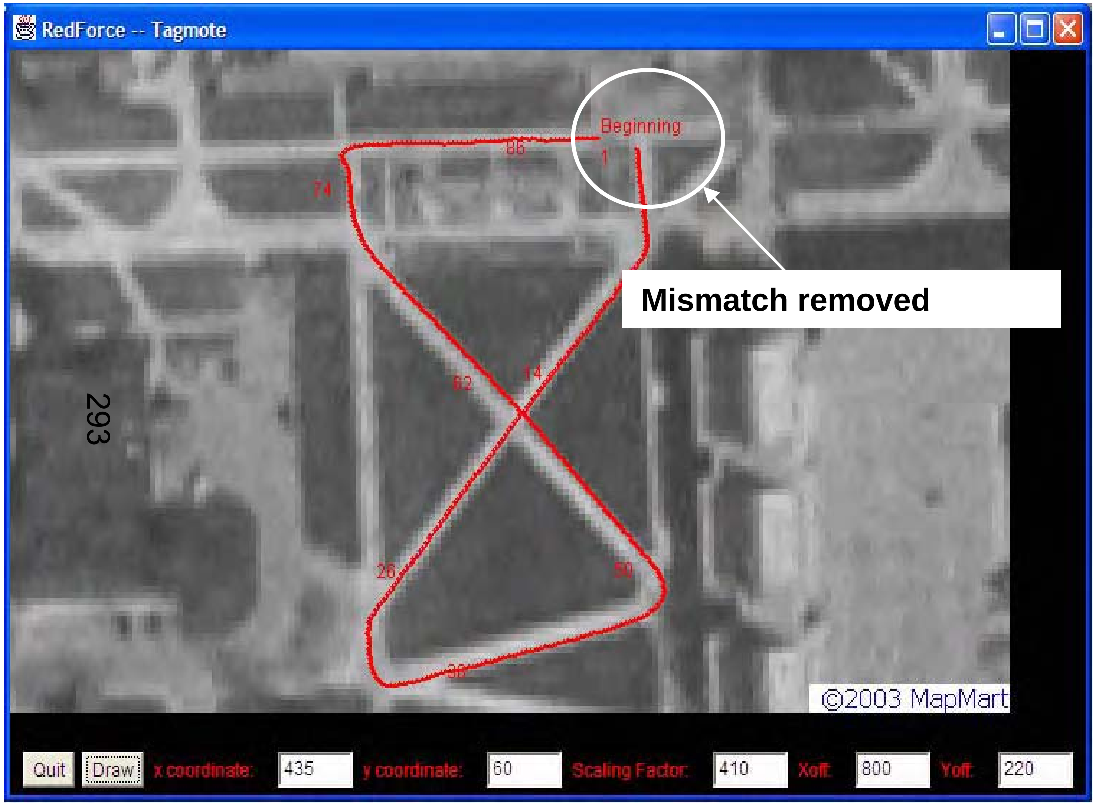
Heading Calibration





- Calibration is essential in order to obtain good performance
- NEST middleware provides a method for in situ calibration of MICA sensors.
- Experimental heading error: 6.8° without calibration, 1.1° after dynamic and static calibration
- NEST localization services can be used to provide “absolute” position fixes that can be transmitted to TagMote.

Post Reconstruction Analysis



RedForce -- Tagmote

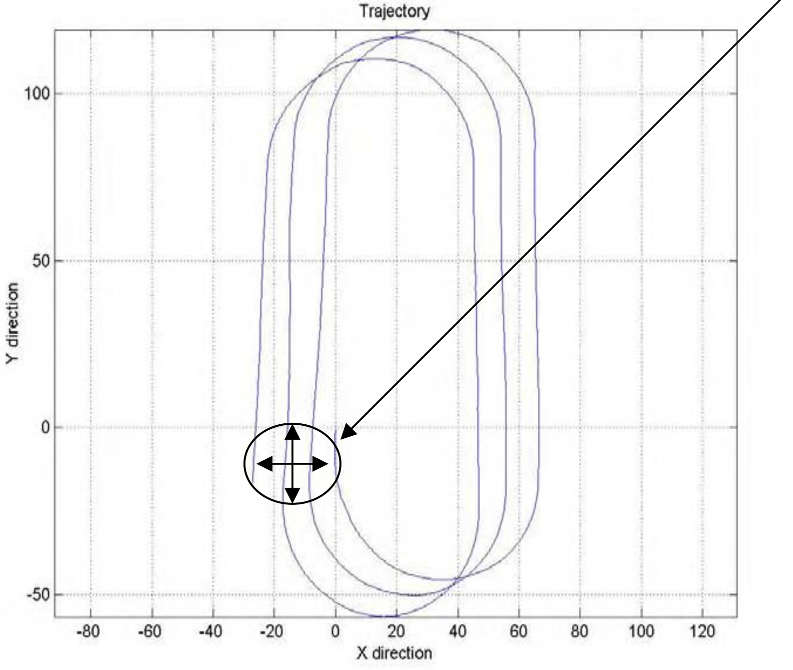
Quit | Draw | x coordinate: 435 | y coordinate: 60 | Scaling Factor: 410 | Xoff: 800 | Yoff: 220

Mismatch removed

- Calibration is essential to obtain good performance
  - Dynamic (Tilt) Calibration
  - Static Calibration
- Calibration constants, however, are never exact.
- Post Reconstruction Analysis fits exfiled data to known waypoints.

Overall Performance

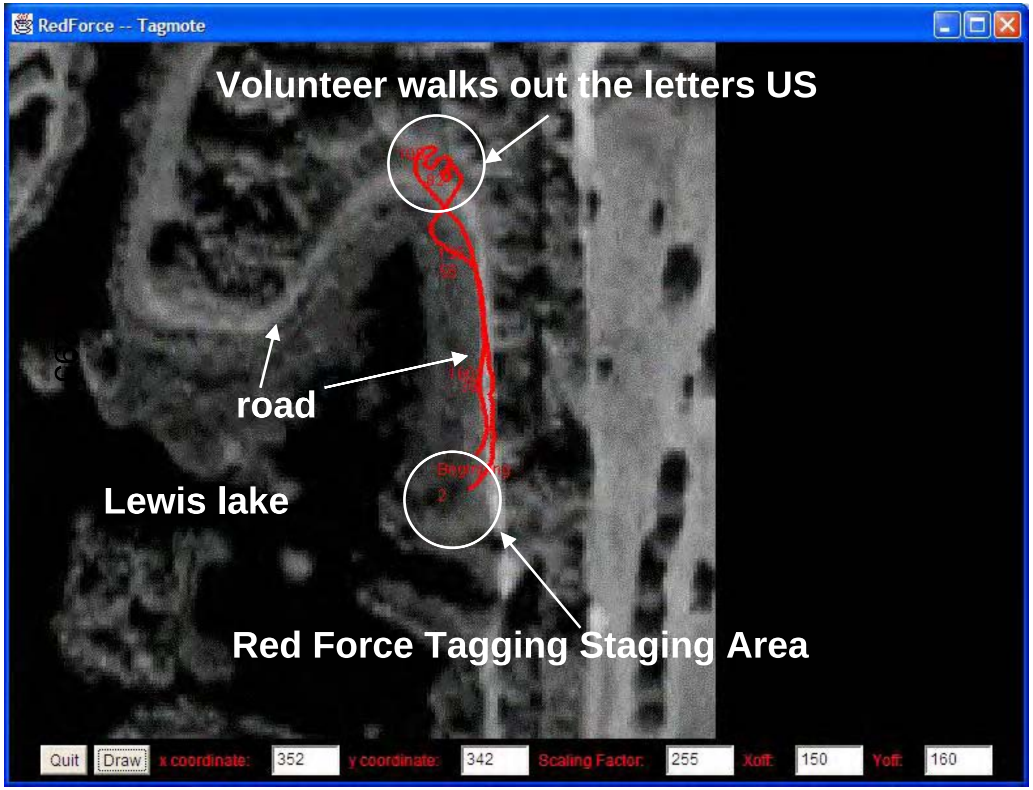
1.2 km walk around track



- Distance Accuracy: ±3% for same subject
- Heading Accuracy: ±1°
  - Example: (27m,16m) error over 1200-meter walk

- Download time
  - 3 min. walk = 3 min. download
  - 70 bytes/sec (data)
- Exfiltration time
  - 3 min. walk = 1 min. exfiltrate
  - 200 bytes/sec (data)
- Reliability: 90-100 %

Performance-Field Experiment



Volunteer walks out the letters US

Red Force Tagging Staging Area

x coordinate: 352 y coordinate: 342 Scaling Factor: 255 Xoff: 150 Yoff: 160

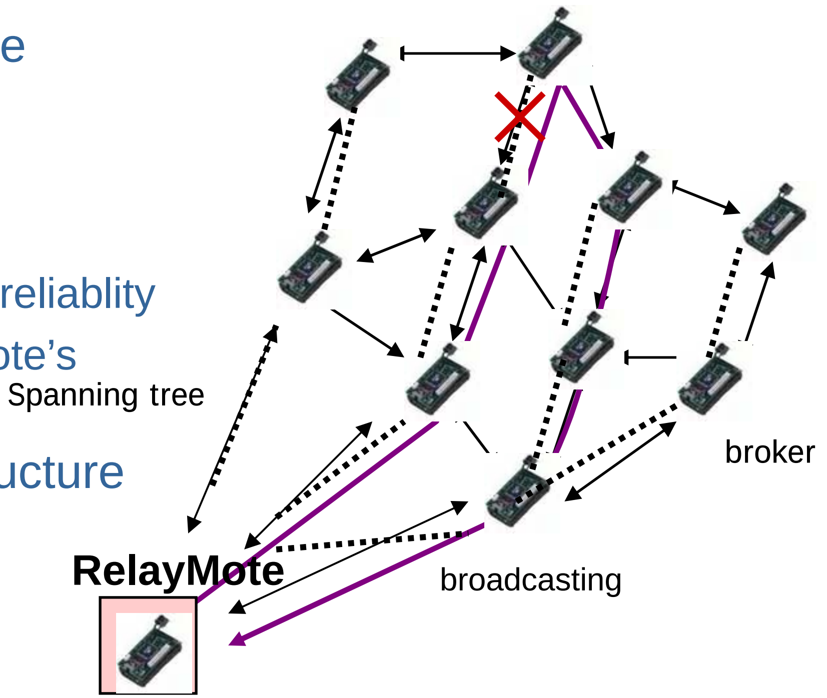
Download trajectory data to NEST, exfiltrate the data to the Relay, reconstruct and display the subject's trajectory.

Outcome:

- 100 percent packet recovery
- 20 meter accuracy over 1 km run
- 3 minute download time/ 3 minute walk (70 data bytes/sec)
- 1 minute exfiltration time/ 3 minute walk (200 data bytes/sec)

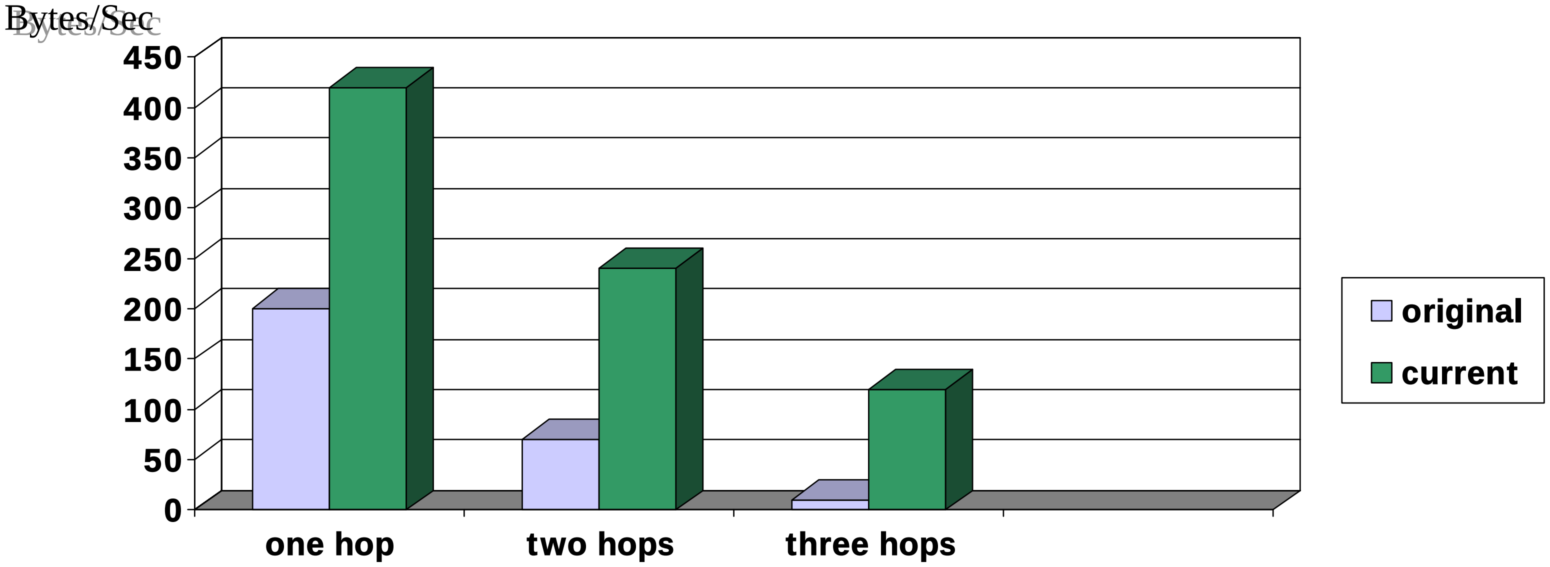
Enhanced Backbone Formation Service

- Red Force Tagging (earlier) Performance
  - – Limited Connectivity in Spanning Tree
    - only kept a single next-hop in the tree
- Enhanced Backbone service
  - – Keeps track of “retries” as estimate of link reliablity
  - – Keeps track of multiple “next hops” in a mote’s neighborhood
- Provided a highly connected routing structure upon which to exfiltrate TagMote data



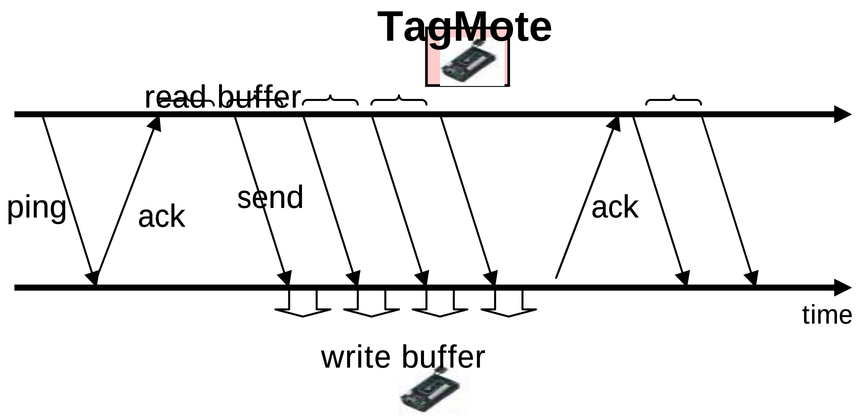
Enhanced Exfiltration Service

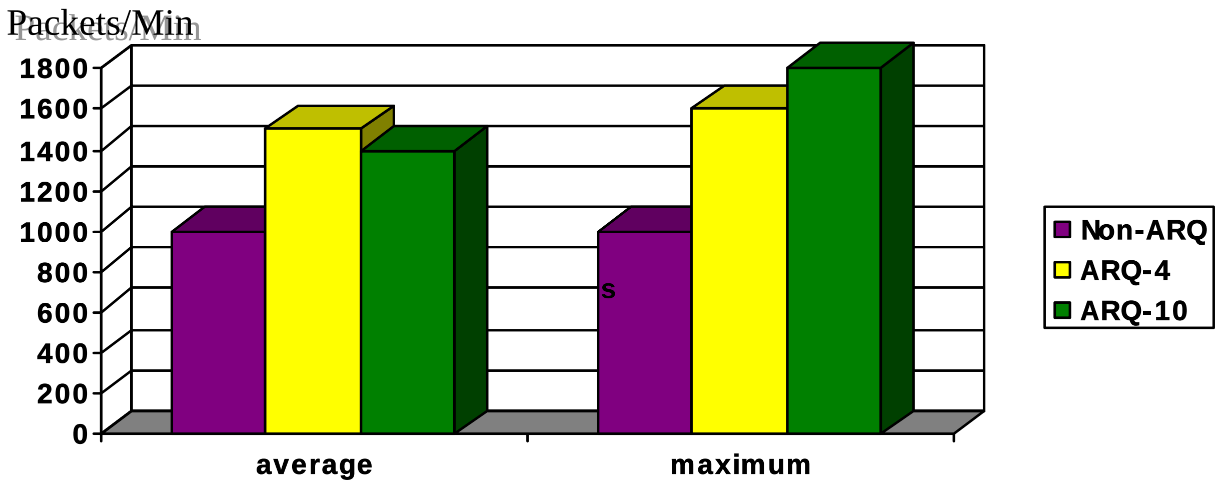
- Red Force Tagging (earlier) Performance
  - Low throughput caused by limited connectivity in spanning tree
  - Deadlocks due to missed messages
- Enhanced Exfiltration Service
  - Switches next-hop based on throughput estimate
  - Watchdog timer to reset “dead” motes in a basic service cell
- Performance of Enhanced Service
  - linear scaling of throughput with number hops
  - higher throughput
  - multiple tests on a 50-node NEST field



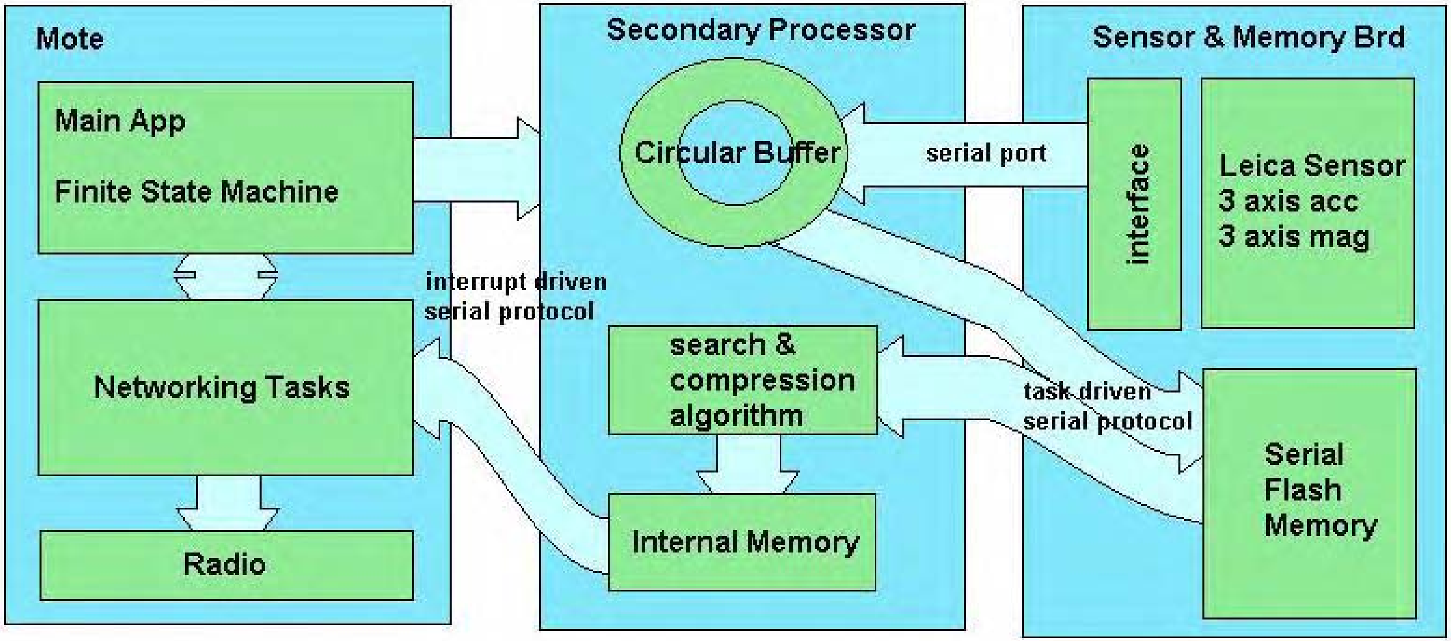
Enhanced Download Service

- Red Force Tagging (Florida) Performance
  - – Slow download rates
- Enhanced Download Service
  - – ARQ-N scheme
- Enhanced Performance
  - – field tested
  - –improved throughput





New HW Design-Adding Microprocessor



Additional TagMote HW Considerations

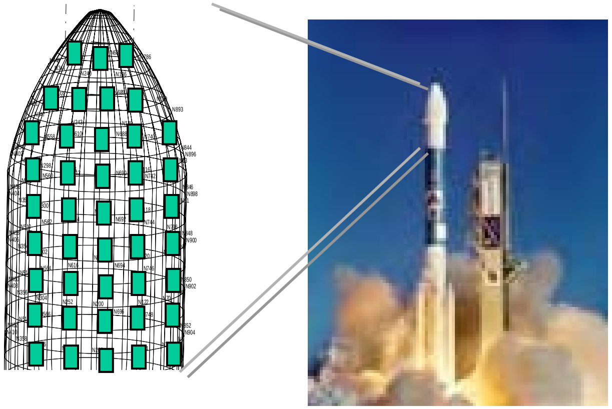
- Programmable Logic
  - future, reduction of production costs
- Redesign the TagMote without the Leica sensor to reduce size and costs

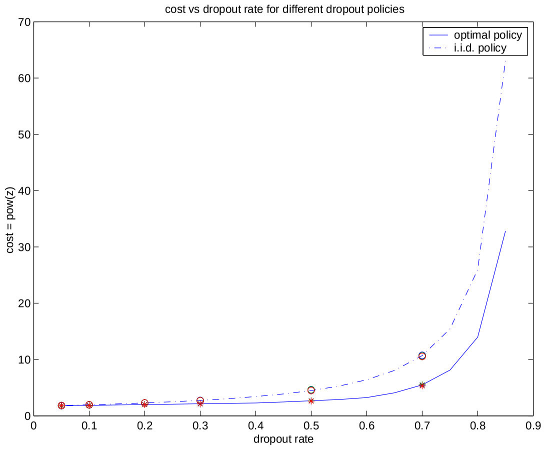
Outline

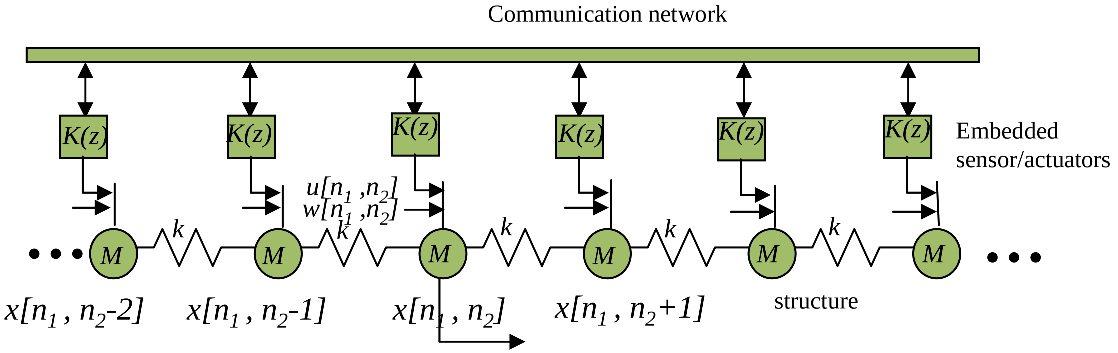
- **Red Force Tagging**
- **Overload Management in Sensor/Actuator Networks**
  - Relationship between application performance and network Quality-of-Service
  - Also studied effect of networking on stability and performance. Distributed dynamical systems
- **Large Networks of Embedded Sensors**
- **Program Issues**

Overload Management

- These methods were applied to a modified version of the Boeing OEP problem
- Localized Spatially Distributed Controller Used
- Examined “overload management” scheme that purposely drops messages to assure all controllers have fair access to the network.
- Analytical way to determine “optimal” dropout sequences for distributed systems.



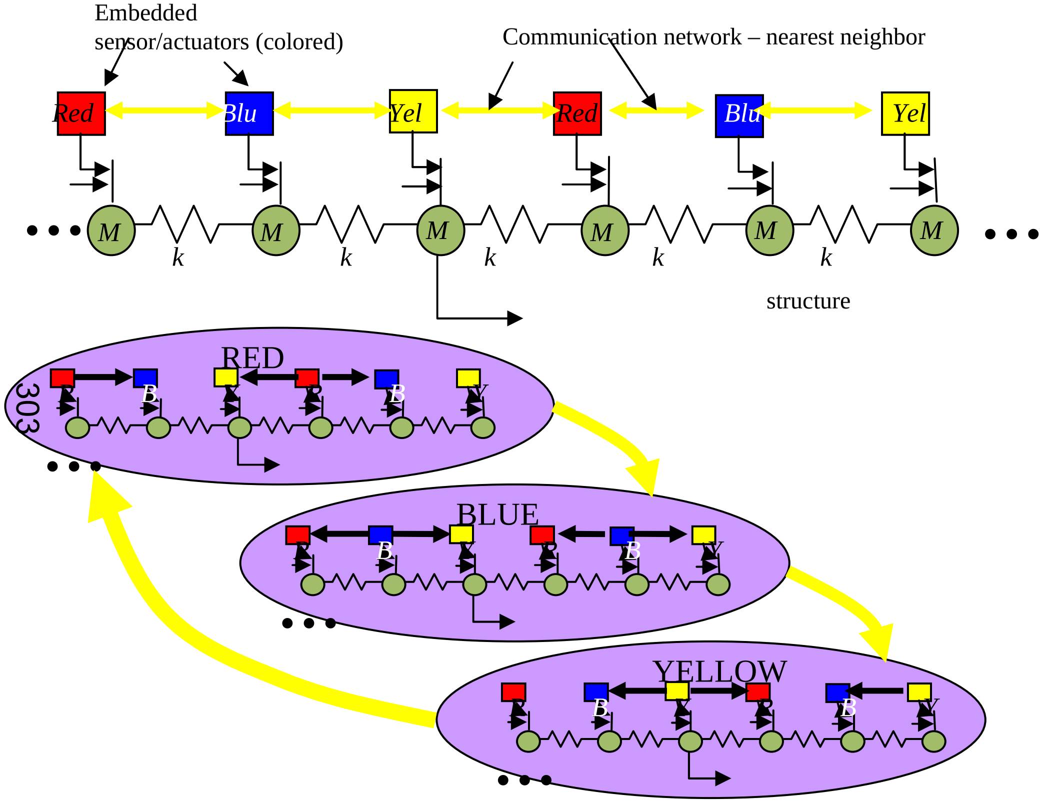


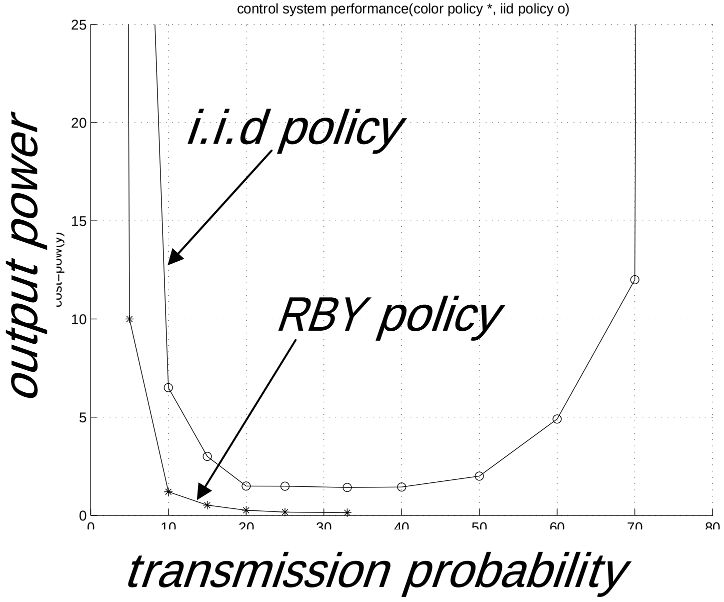


[4] M. D. Lemmon, Q. Ling and Y. Sun (2003) “Overload management in sensor-actuator networks used for spatially distributed control systems”, 1st ACM conference on Sensor Networks (SenSys03), UCLA 2003.

Cross Layer Control over Sensor-Actuator Networks

- Colored Network Abstraction
- Distributed Jump Linear system model
- Power versus scheduling policy


Embedded
sensor/actuators (colored)
Communication network – nearest neighbor
Red
Blu
Yel
Red
Blu
Yel
M
k
structure
RED
BLUE
YELLOW
R
B
Y



Cross layer approach uses relationships between application performance and network QoS to design scalable systems.

Outline

- Red Force Tagging
- Overload Management in Sensor/Actuator Networks
- Large Networks of Embedded Sensors
  - Power consumption and throughput in regular and random networks
- Program Issues

Distances in Random Networks

The distance R_n to the n-th neighbor in an m-dimensional random network has the pdf (generalized Weibull distribution):

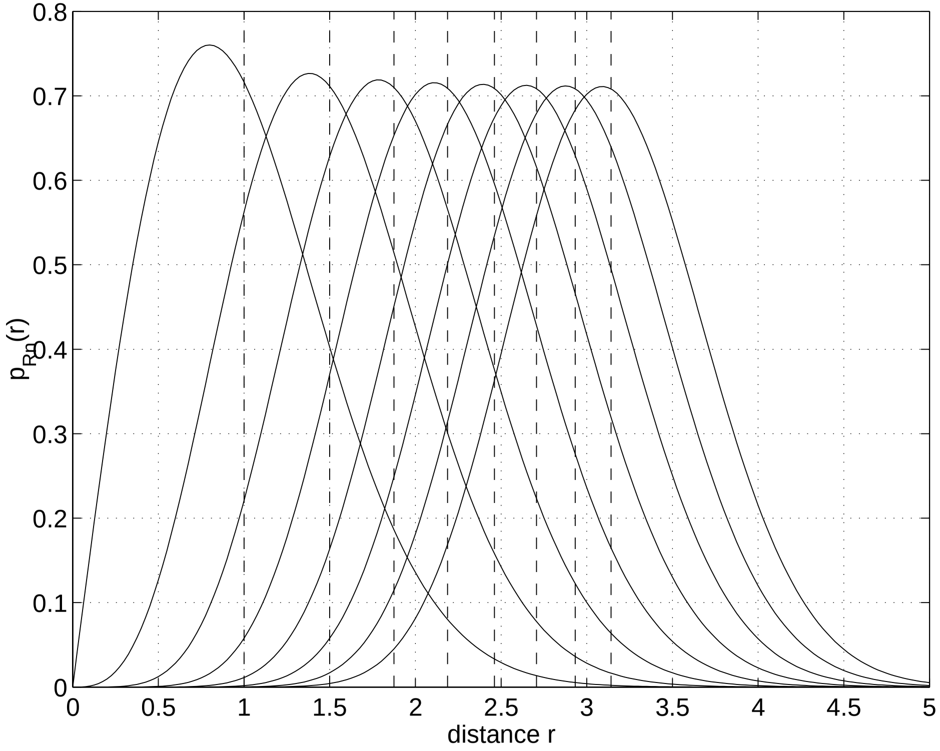
$$f_{R_n}(r) = e^{-\lambda c_m r^m} \frac{m\left(\lambda c_m r^m\right)^n}{r(n-1)!},$$

Where $c_m r^m$ is the volume of the m-sphere. The cdf can be written as:

$$F_{R_n}(r) = 1 - \frac{\Gamma_{\text{ic}}(n, \lambda c_m r^m)}{\Gamma(n)},$$

Where Γ_{ic} is the incomplete Γ function.

In two-dimensional networks, the mean increases with $n^{1/2}$, and the variance is (almost) independent of n.



Expected Distance and Energy Consumption

The expected distance to the nearest neighbor in a sector φ is

$$\mathbb{E}[R_n] = \left(\frac{1}{\lambda c_{\phi,m}}\right)^{\frac{1}{m}} \frac{\Gamma\left(n + \frac{1}{m}\right)}{\Gamma(n)}.$$

With a path loss exponent α, the expected energy consumption is proportional to

$$\mathbb{E}[R_n^\alpha] = \left(\frac{1}{\lambda c_{\phi,m}}\right)^{\frac{\alpha}{m}} \frac{\Gamma\left(n + \frac{\alpha}{m}\right)}{\Gamma(n)},$$

which is convex in n since α>m is required for the interference to remain finite.

Comparison of Regular and Random Networks

One-dimensional networks

	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
g_{ave} of regular line network	0.0538	0.0875	0.1138	0.1327
g_{ave} of random line network	0.0495	0.0811	0.1080	0.1266
g_{min} of regular line network	0.0576	0.0855	0.1078	0.1231
g_{min} of random line network	0.0352	0.0503	0.0614	0.0669
E_{ave} of regular line network	1	1	1	1
E_{ave} of random line network	2	6	24	120
E_{max} of regular line network	1	1	1	1
E_{max} of random line network	14.5	66.2	340.2	1974.5

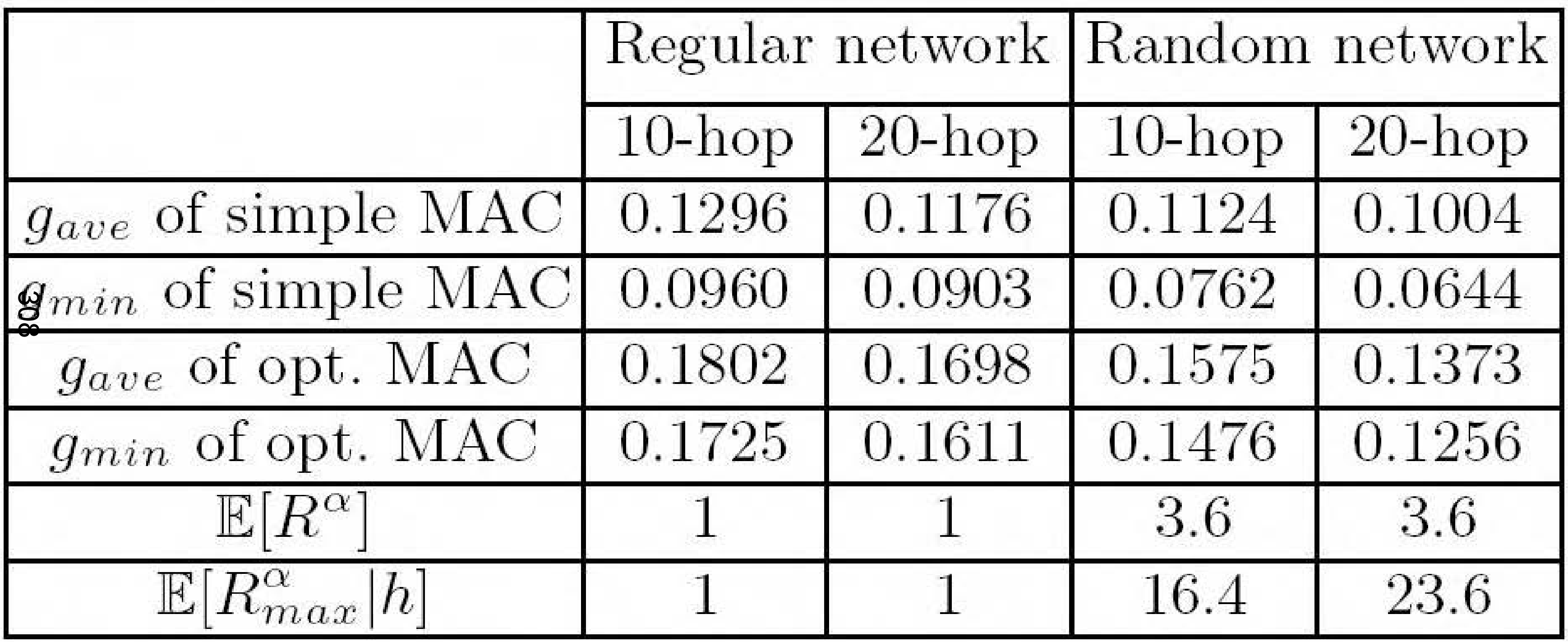
The maximum is taken over k=20 hops:

$$\mathbb{E}[\max\{E_1, \ldots, E_k\}] = \int_0^\infty (1 - F_E(r)^k)dr\,,$$

where F_E is the cdf of R_n^α.

Comparison of Regular and Random Networks

Two-dimensional networks



	Regular network		Random network	
	10-hop	20-hop	10-hop	20-hop
g_{ave} of simple MAC	0.1296	0.1176	0.1124	0.1004
g_{min} of simple MAC	0.0960	0.0903	0.0762	0.0644
g_{ave} of opt. MAC	0.1802	0.1698	0.1575	0.1373
g_{min} of opt. MAC	0.1725	0.1611	0.1476	0.1256
$\mathbb{E}[R^\alpha]$	1	1	3.6	3.6
$\mathbb{E}[R^\alpha_{max}\|h]$	1	1	16.4	23.6

h denotes the number of hops (10, 20).
The path loss exponent α=4.

Routing over Non-nearest Neighbors

Solution: Transmit to the furthest neighbor that can be reached with high enough probability.

Pdf of the distance to the furthest neighbor within distance d, (given that there is at least one node within d):

$$f_{R_d}(r) = \frac{\lambda c_{\phi,m} m r^{m-1} e^{\lambda c_{\phi,m} r^m}}{e^{\lambda c_{\phi,m} d^m} - 1}, \quad r \in [0, d].$$

As d grows, the performance of the random network approaches the performance of the regular network.

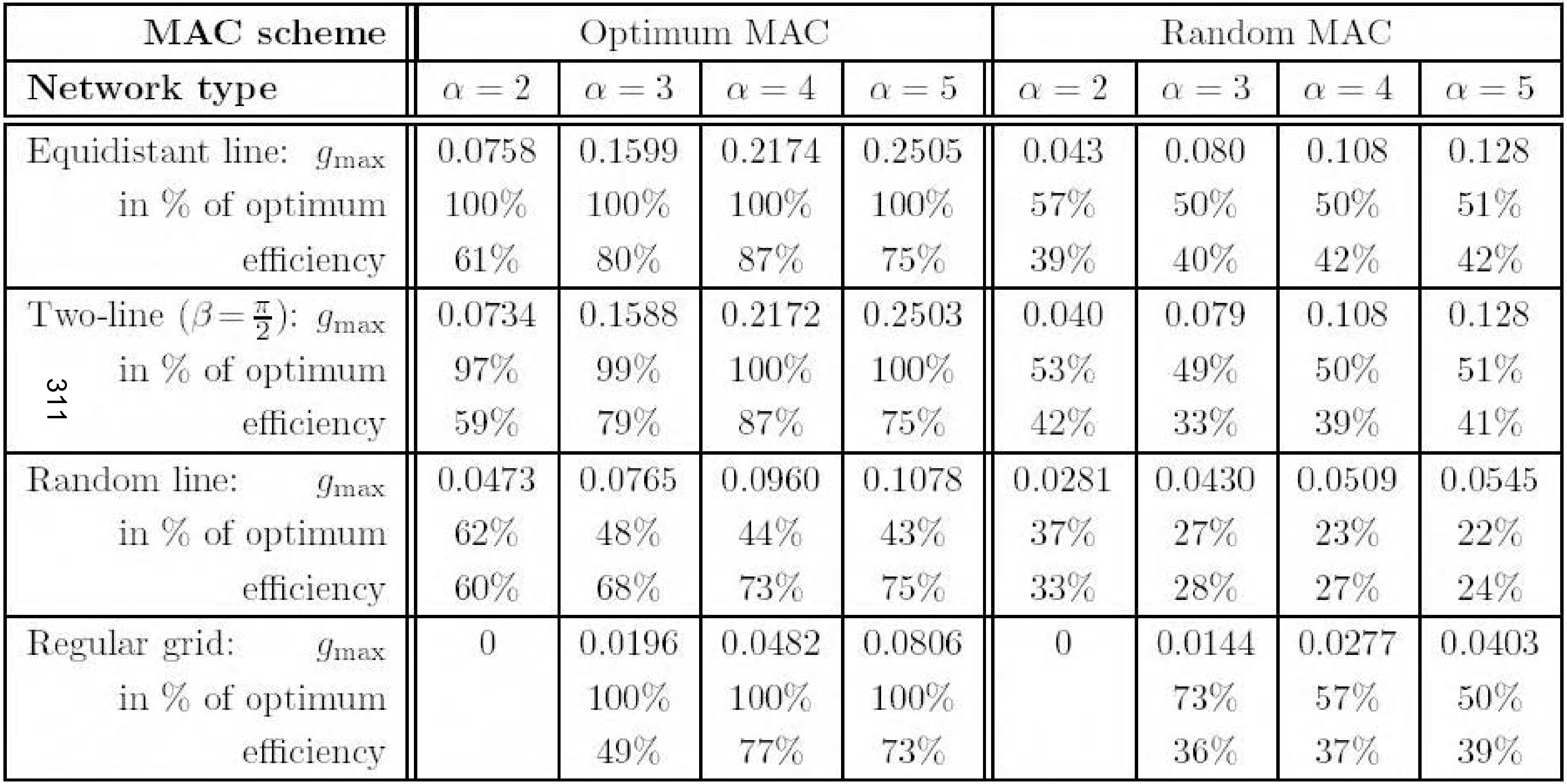
Separation Property of Rayleigh Fading Channels

For Rayleigh fading channels, packet reception probabilities can be factorized into a noise part and an interference part:

$$
\begin{aligned}
p_r =& \mathbb{P}[R_0 \geqslant \Theta(I+N)] \\
=& \exp\left(-\frac{\Theta(I+N)}{\bar{R}_0}\right) \\
=& \int_0^\infty \cdots \int_0^\infty \exp\left(-\frac{\Theta(\sum_{i=1}^k r_i + N)}{\bar{R}_0}\right) \prod_{i=1}^k p_{R_i}(r_i)\,\mathrm{d}r_1 \cdots \mathrm{d}r_k \\
=& \underbrace{\exp\left(-\frac{\Theta N}{P_0 d_0^{-\alpha}}\right)}_{p_r^N} \cdot \underbrace{\prod_{i=1}^k \frac{1}{1+\Theta \frac{P_i}{P_0}\left(\frac{d_0}{d_i}\right)^\alpha}}_{p_r^I} .
\end{aligned}
$$

Note that this turns into a “disk model” for α→∞.

Achievable Throughput in Fading Networks



MAC scheme	Optimum MAC				Random MAC			
Network type	$\alpha=2$	$\alpha=3$	$\alpha=4$	$\alpha=5$	$\alpha=2$	$\alpha=3$	$\alpha=4$	$\alpha=5$
Equidistant line: $g_{\max}$	0.0758	0.1599	0.2174	0.2505	0.043	0.080	0.108	0.128
in % of optimum	100%	100%	100%	100%	57%	50%	50%	51%
efficiency	61%	80%	87%	75%	39%	40%	42%	42%
Two-line $(\beta=\frac{\pi}{2})$: $g_{\max}$	0.0734	0.1588	0.2172	0.2503	0.040	0.079	0.108	0.128
in % of optimum	97%	99%	100%	100%	53%	49%	50%	51%
efficiency	59%	79%	87%	75%	42%	33%	39%	41%
Random line: $g_{\max}$	0.0473	0.0765	0.0960	0.1078	0.0281	0.0430	0.0509	0.0545
in % of optimum	62%	48%	44%	43%	37%	27%	23%	22%
efficiency	60%	68%	73%	75%	33%	28%	27%	24%
Regular grid: $g_{\max}$	0	0.0196	0.0482	0.0806	0	0.0144	0.0277	0.0403
in % of optimum		100%	100%	100%		73%	57%	50%
efficiency		49%	77%	73%		36%	37%	39%

The difference between the simplest MAC scheme and the optimum MAC scheme is only a factor 2.

(M. Haenggi and Xiaowen Liu, “*Fundamental Throughput Limits in Rayleigh Fading Sensor Networks*”, submitted to IEEE Journ. On Sel. Areas in Comm.)

Conclusions for Network Design

- In random networks, nearest-neighbor communication is very inefficient.
  Power amplifiers need a high Pmax to ensure connectivity. This maximum power should be used at every hop, the higher it is, the better.
  This does not hurt the throughput, but it achieves energy balancing among nodes, reduces delay, and provides redundancy in the case of node failures.

- Related to the MAC scheme, a lot of effort (time, infrastructure traffic) is required to achieve the level of coordination needed for good MAC schemes (low collision probability).
  The achievable gain is only 2, so it may not be worth it.
  Make sure the data rates are high enough such that the traffic is light. Then simple random MACs are near-optimum.

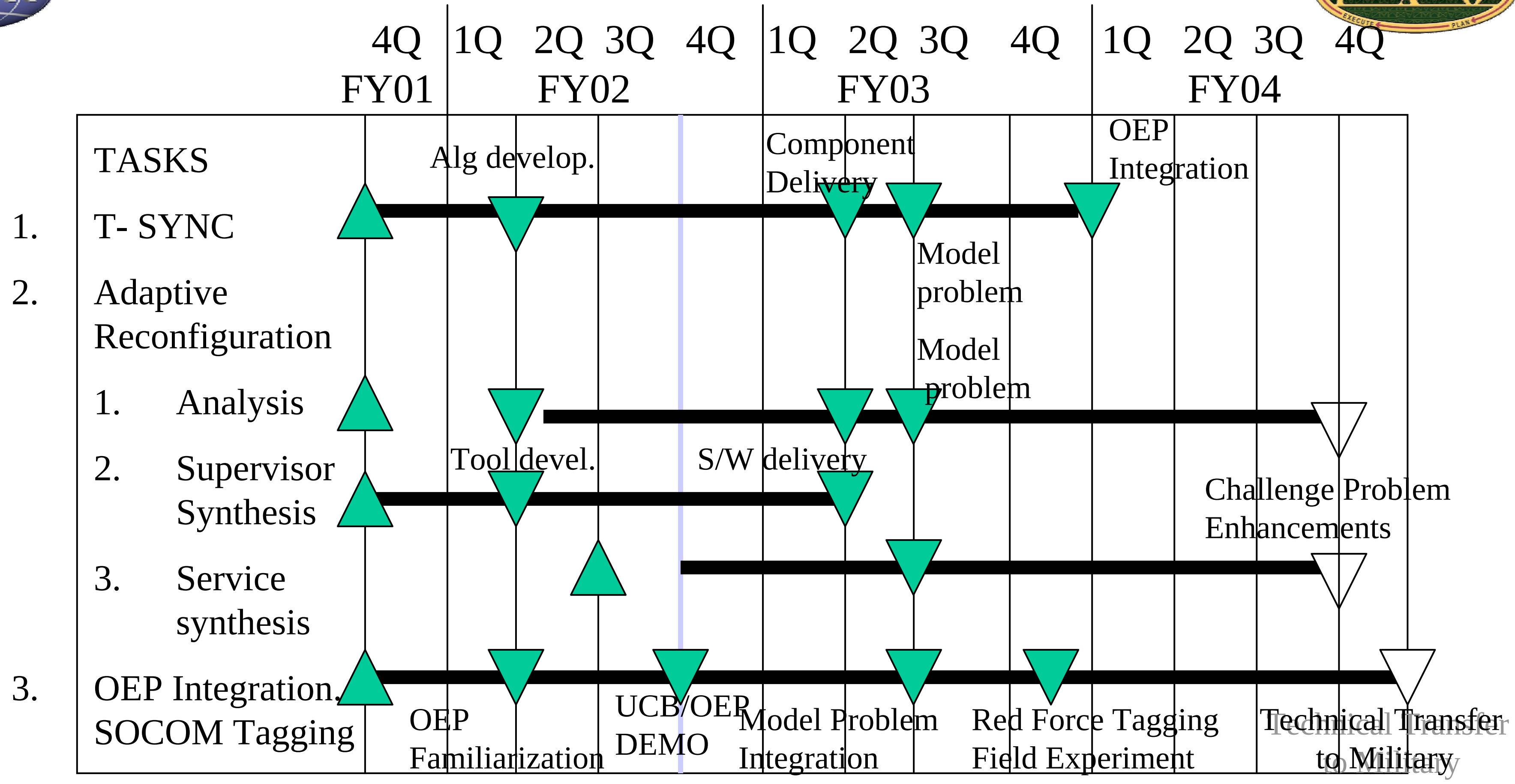
Outline

- Red Force Tagging
- Overload Management in Sensor/Actuator Networks
- Large Networks of Embedded Sensors
- Program Issues
  - Milestones, future plans, issues

Project Plans

- Plan over next 6 months
  - Red Force Tagging
    - Extensions from demo to stage closer to ConOps that involve speed of download, network discovery, security, TagMote and network middleware interaction
  - NEST Services
    - Improve download time for current services
    - Improve robustness and scalability of middleware services
- Success Goals
  - TagMote to NEST download = (1 min walk to 15 seconds of download time

Project Schedule and Milestones



- On schedule to meet FY04 milestones.
- SOCOM Challenge Problem Enhancements.
- Robust TagMote and Middleware services. Wireless Networks-power issues

Technical Transition/Transfer

- Transition Activities
  - SOCOM
- Opportunities
  - Honeywell-initiated discussions
  - Universities

Program Issues

- Communication issues in Large scale networks:

It is known that ad hoc networks with random source-destination pairs scale badly.
It is crucial to identify communication schemes, architectures, and traffic patterns that overcome these limitations.

Red Force Tagging Field Experiment

MacDill AFB, August 20, 2003

P.J. Antsaklis, Martin Haenggi, M.D. Lemmon
Dept. of Electrical Engineering
University of Notre Dame

DARPA/IXO NEST PI MEETING
December 2003, Santa Fe NM

Personnel

Senior Personnel

Panos J. Antsaklis, Michael D. Lemmon, Martin Haenggi

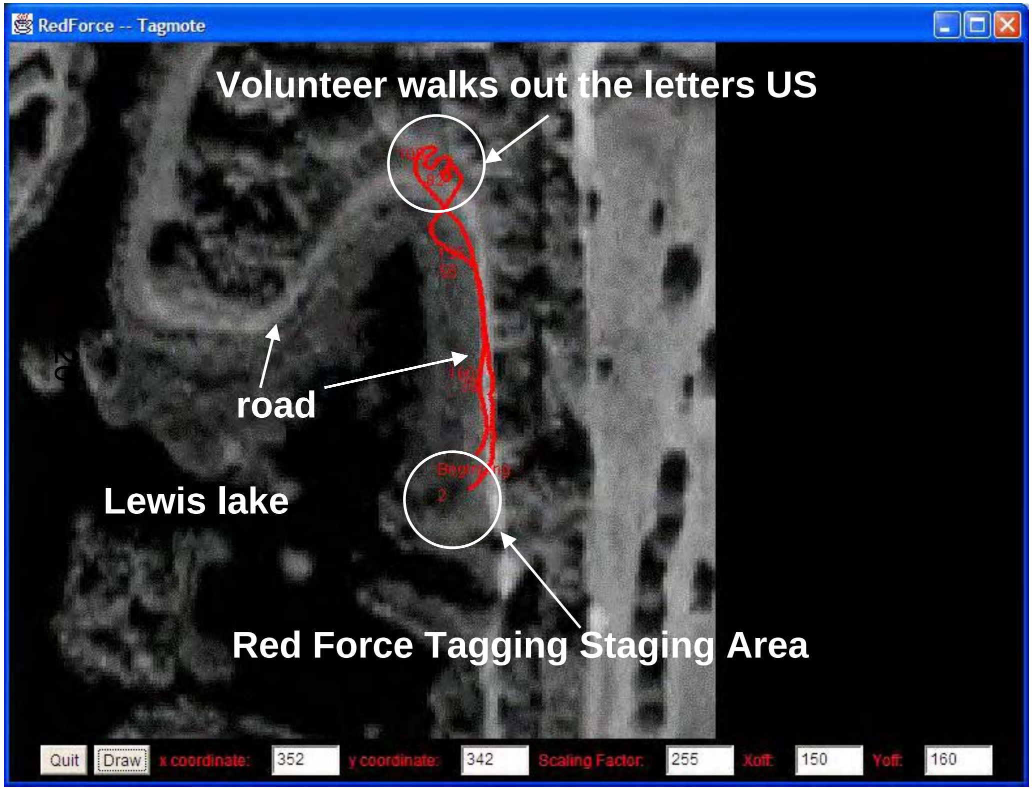
Electrical Engineering, University of Notre Dame

Graduate Students

Luis Montestruque, Brett McMickell, Yashan Sun, Lei Fang

Hui Fang, Ioannis Koutroulis

Performance



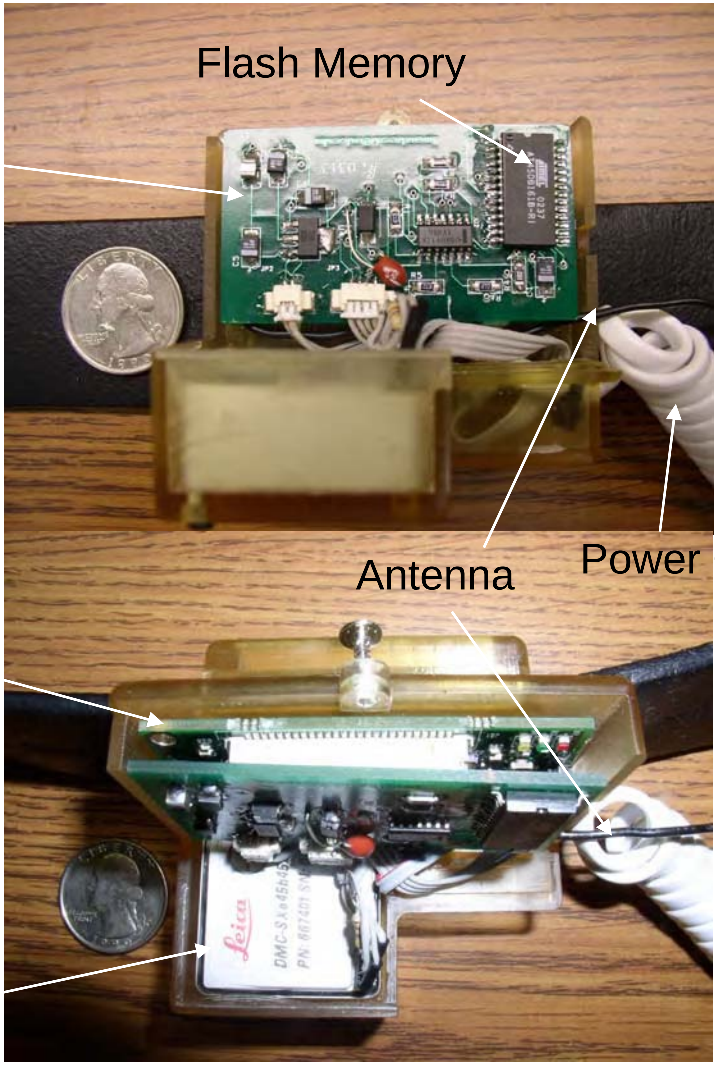
Download trajectory data to NEST, exfiltrate the data to the Relay, reconstruct and display the subject's trajectory.

Outcome:

- 100 percent packet recovery
- 20 meter accuracy over 1 km run
- 3 minute download time/ 3 minute walk (70 data bytes/sec)
- 1 minute exfiltration time/ 3 minute walk (200 data bytes/sec)

TagMote: Current Design

- Compact: 65×58×24 mm
- Lightweight:100 grams



Flash Memory

Tagmote Board

Antenna

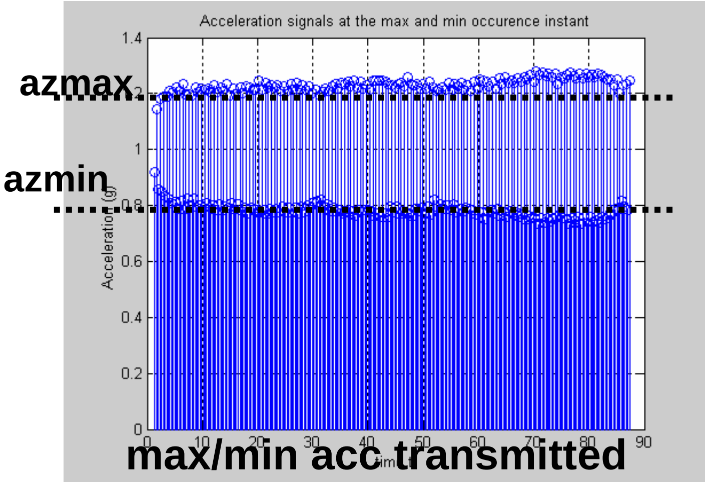
Power Cord

MICA Board

Leica DMC-SX

RED FORCE TAGGING

TagMote Processing

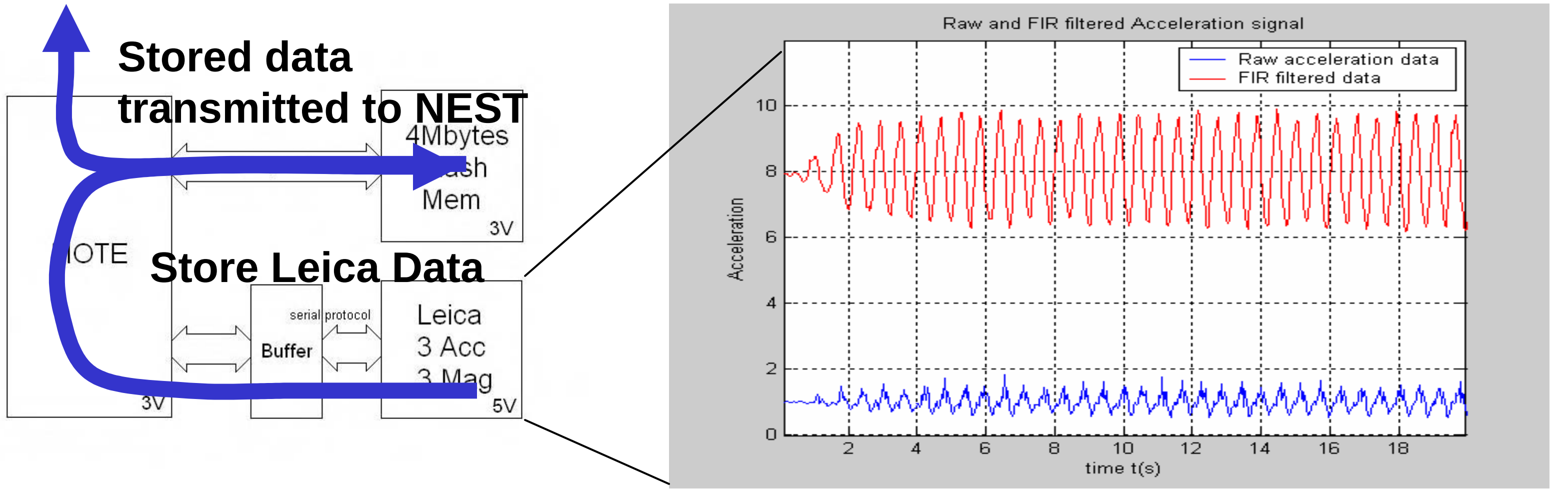


azmax

azmin

max/min acc transmitted

Raw accelerometer data is smoothed and downsampled (30 Hz to 4 Hz) at TagMote



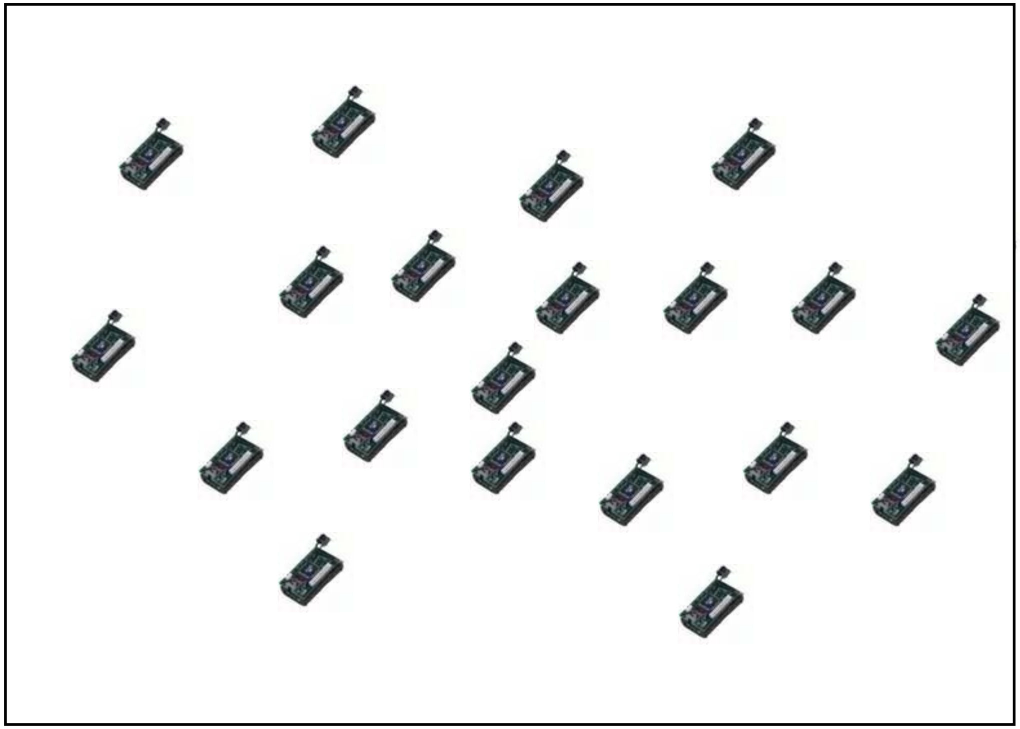
Stored data transmitted to NEST

Store Leica Data

Technical Approach – NEST Middleware

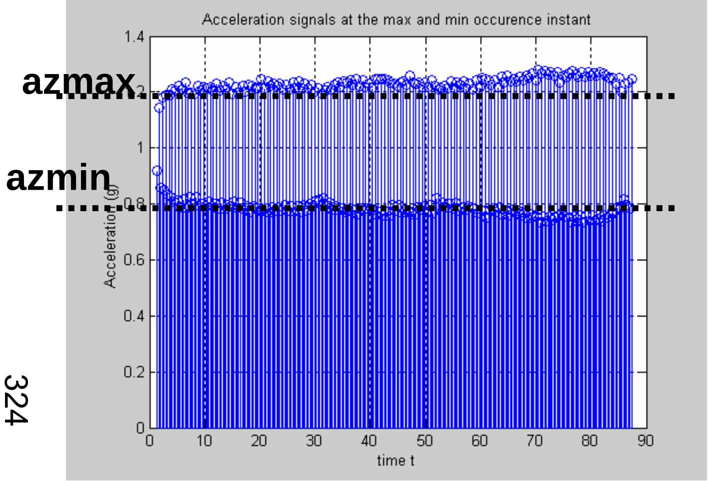
- NEST Middleware Challenges
  - Download large amounts of data from TagMote to NEST
  - Exfiltrate large amounts of NEST data to SOCOM relay
  - Reliable and Timely Data Delivery
- Solutions
  - Multi-hop Streaming service (link reservation)
  - Explicit packet acknowledgement
  - Route reconfiguration (based on link level timeouts)

Clear Basic Service Cell



Step Length Estimation

Exfiltrated acc. data



Exfiltrated azmax/azmin data is processed at Relay Station to extract

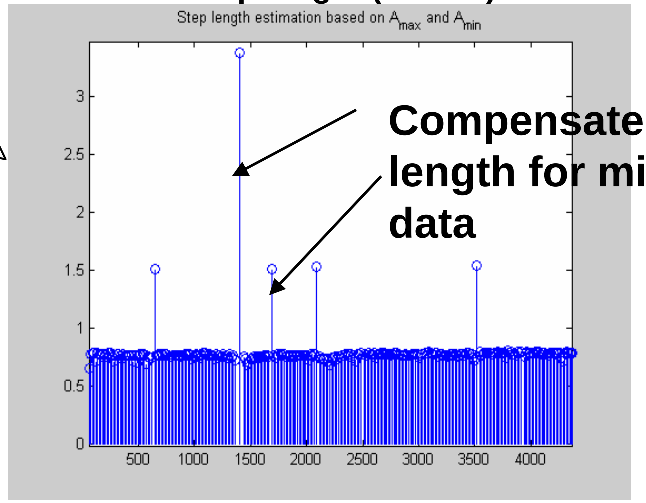
- number of steps
- step length

- **Step length estimation**

$$K \times \sqrt[4]{a_{z\max} - a_{z\min}}$$

K = 0.51 ~ 0.57
for wide range of individuals on different surfaces

Step length (meters)

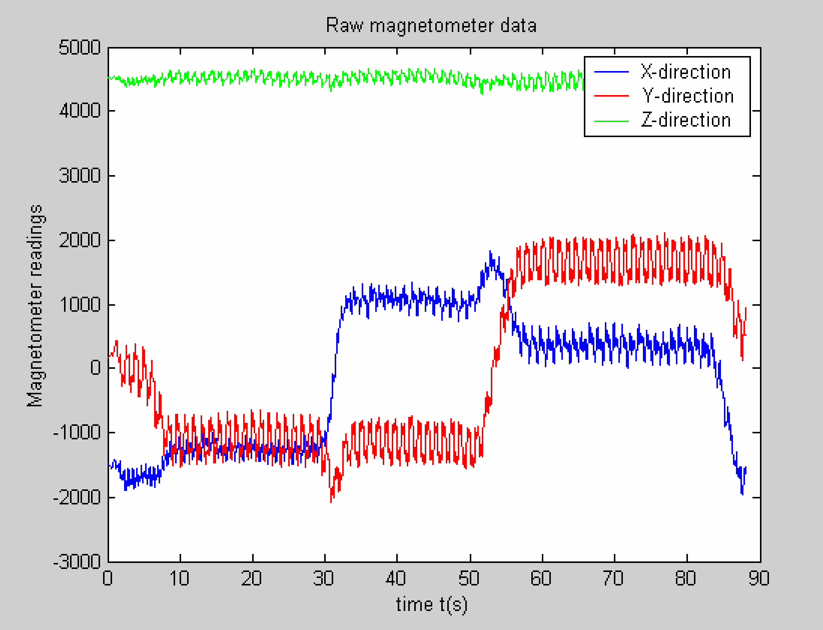


Compensated step length for missing data

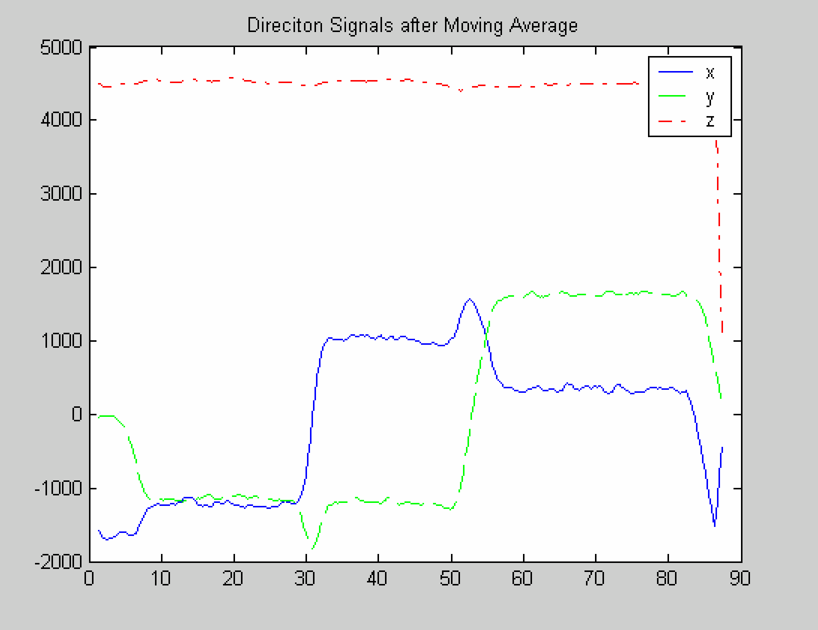
Heading Estimation

- Moving-average

Raw Compass Data



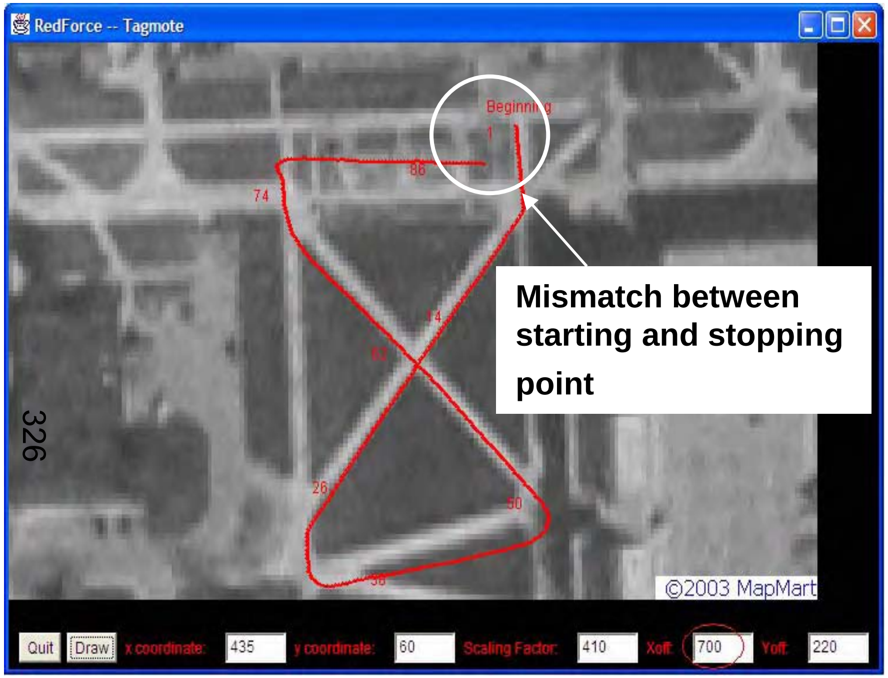
After Moving-average



- Dynamic calibration: Tilt compensation using 3-D accelerometer data
- Static calibration: Magnetic field distortion (e.g., due to human body)

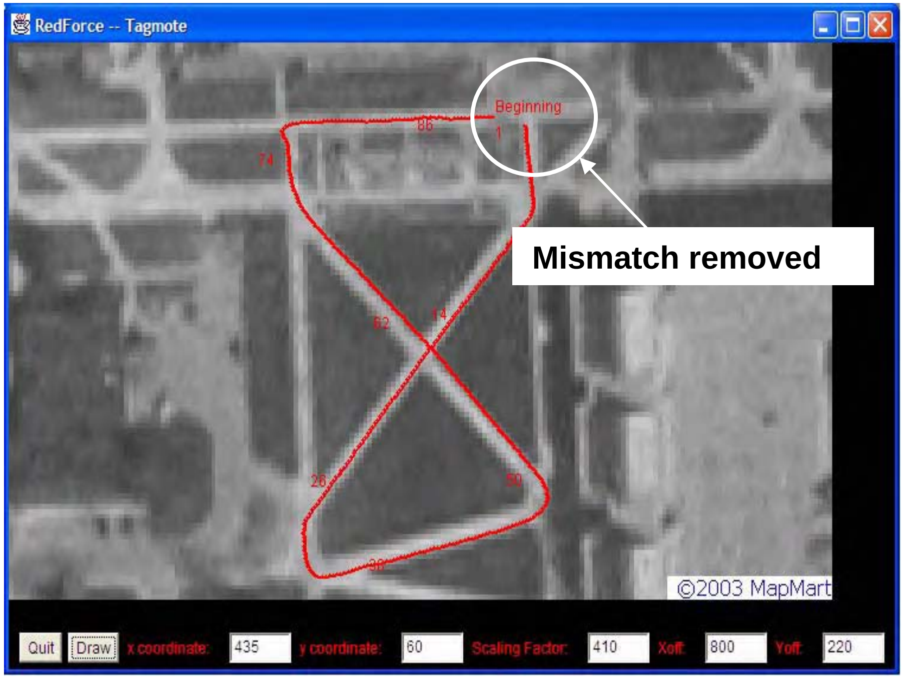
RED FORCE TAGGING

Trajectory Reconstruction GUI



x coordinate: 435 y coordinate: 60 Scaling Factor: 410 Xoff: 700 Yoff: 220

Mismatch between starting and stopping point

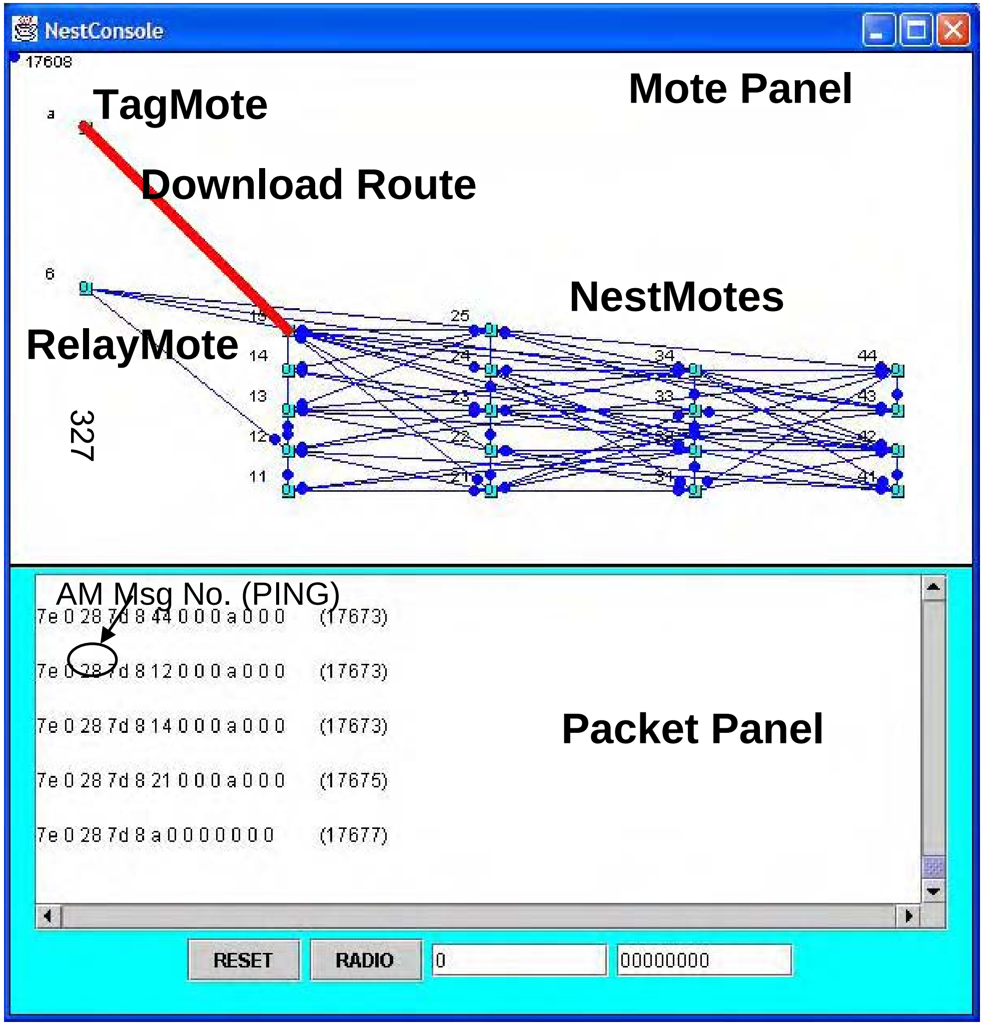


x coordinate: 435 y coordinate: 60 Scaling Factor: 410 Xoff: 800 Yoff: 220

Mismatch removed

- Reconstructed Trajectory Displayed over Aerial Map of Area
- Post-Reconstruction Analysis allows user to adjust trajectory to known landmarks

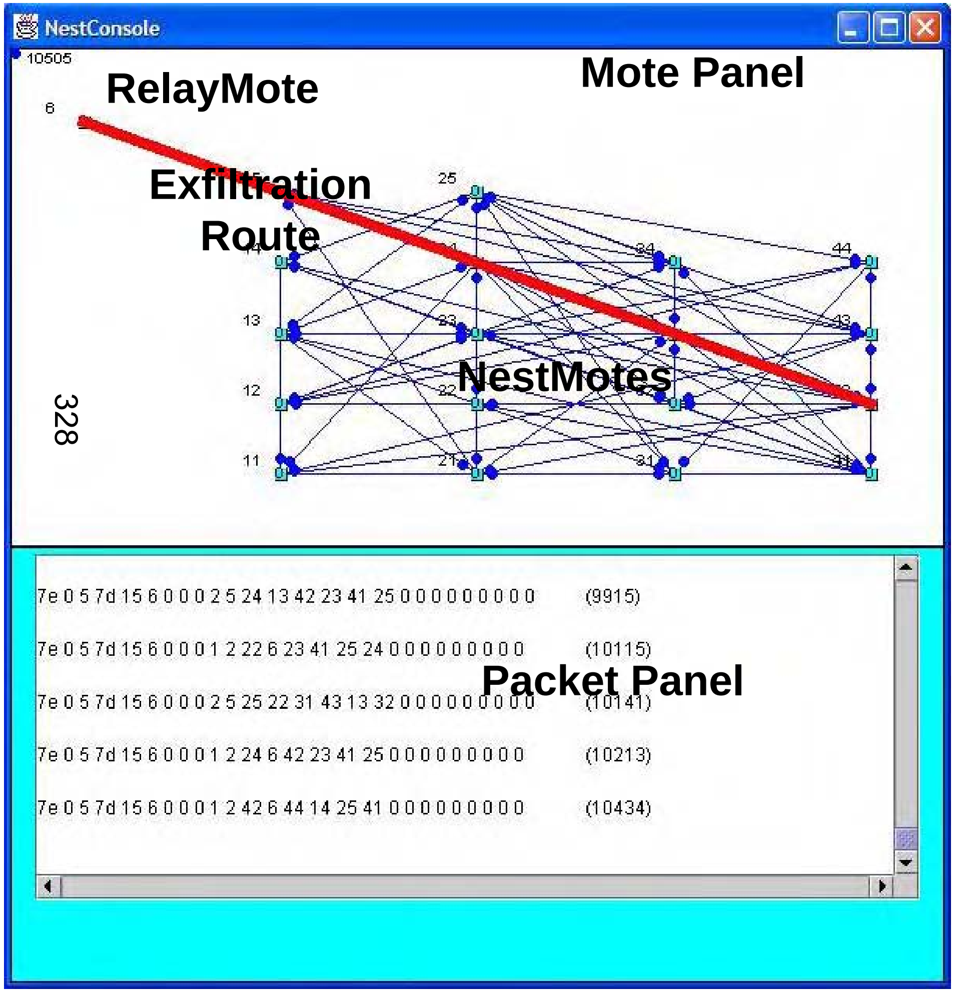
Listener GUI



```
7e 0 28 7d 8 44 0 0 0 a 0 0 0     (17673)

7e 0 28 7d 8 12 0 0 0 a 0 0 0     (17673)

7e 0 28 7d 8 14 0 0 0 a 0 0 0     (17673)

7e 0 28 7d 8 21 0 0 0 a 0 0 0     (17675)

7e 0 28 7d 8 a 0 0 0 0 0 0 0      (17677)
```

- Eavesdrop on network traffic
- Telemetry packets generate graphical display of motes
- Captures data packets downloaded to NEST and stores to file

Relay GUI



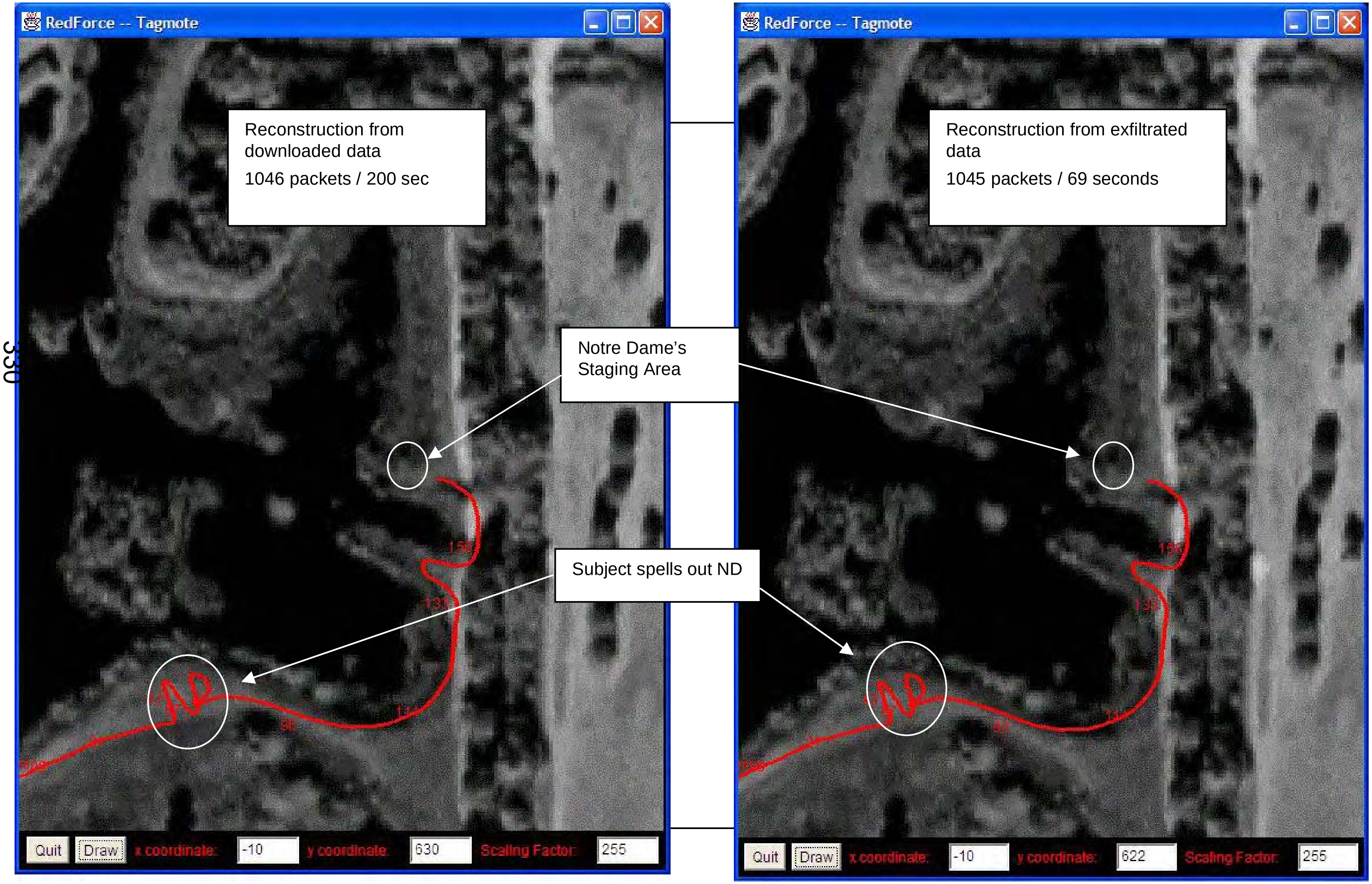
```
7e 0 5 7d 15 6 0 0 0 2 5 24 13 42 23 41 25 0 0 0 0 0 0 0 0 0     (9915)
7e 0 5 7d 15 6 0 0 0 1 2 22 6 23 41 25 24 0 0 0 0 0 0 0 0 0      (10115)
7e 0 5 7d 15 6 0 0 0 2 5 25 22 31 43 13 32 0 0 0 0 0 0 0 0 0     (10141)
7e 0 5 7d 15 6 0 0 0 1 2 24 6 42 23 41 25 0 0 0 0 0 0 0 0 0      (10213)
7e 0 5 7d 15 6 0 0 0 1 2 42 6 44 14 25 41 0 0 0 0 0 0 0 0 0      (10434)
```

- Catches messages sent directly to RelayMote (telemetry/exfile)
- Telemetry packets generate graphical display of NEST
- Exfiltrated Packets saved in data file (acceleration/bearing/time)
- Data file processed to reconstruct trajectory

Field Experiment

	Recovery rate exfiled/download	Download Time (sec)	Download Throughput (bytes/sec)	Exfiltration Time (seconds)	Exfiltration Throughput (bytes/sec)
Trial 1	99.9%	200	84	69	242
Trial 2	103.7%	189	72	111	127
Trial 3	100.0%	189	79	67	222
Trial 4	99.9%	184	63	55	213
Average		190.5	74.5	75.5	201

Experiment 1
(walk from UVA to ND's site)



Reconstruction from downloaded data

1046 packets / 200 sec

x coordinate: -10 y coordinate: 630 Scaling Factor: 255

Reconstruction from exfiltrated data

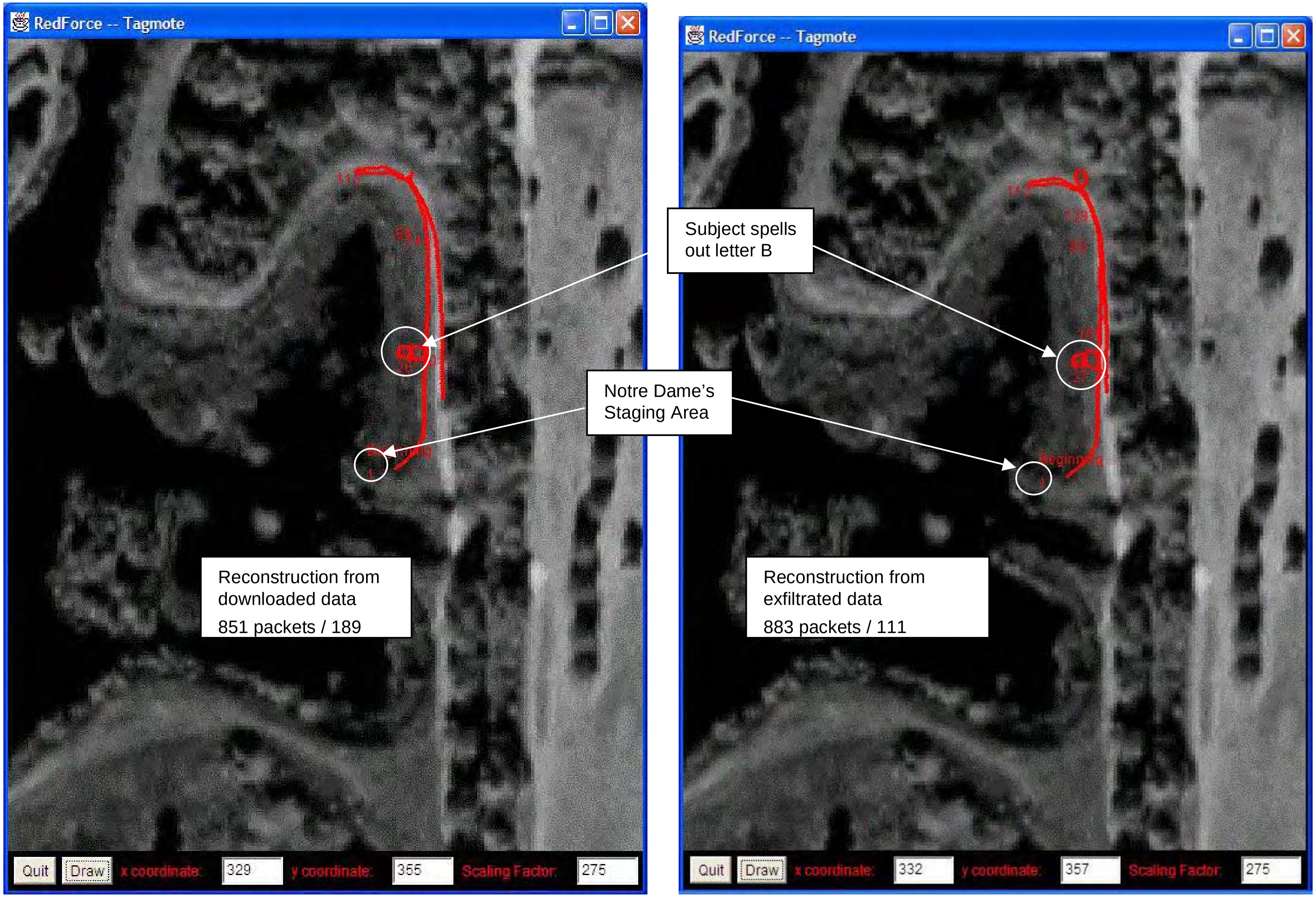
1045 packets / 69 seconds

x coordinate: -10 y coordinate: 622 Scaling Factor: 255

Notre Dame's Staging Area

Subject spells out ND

Experiment 2
(subject's walk north of ND's site)



x coordinate: 329 y coordinate: 355 Scaling Factor: 275

x coordinate: 332 y coordinate: 357 Scaling Factor: 275

Subject spells out letter B

Notre Dame's Staging Area

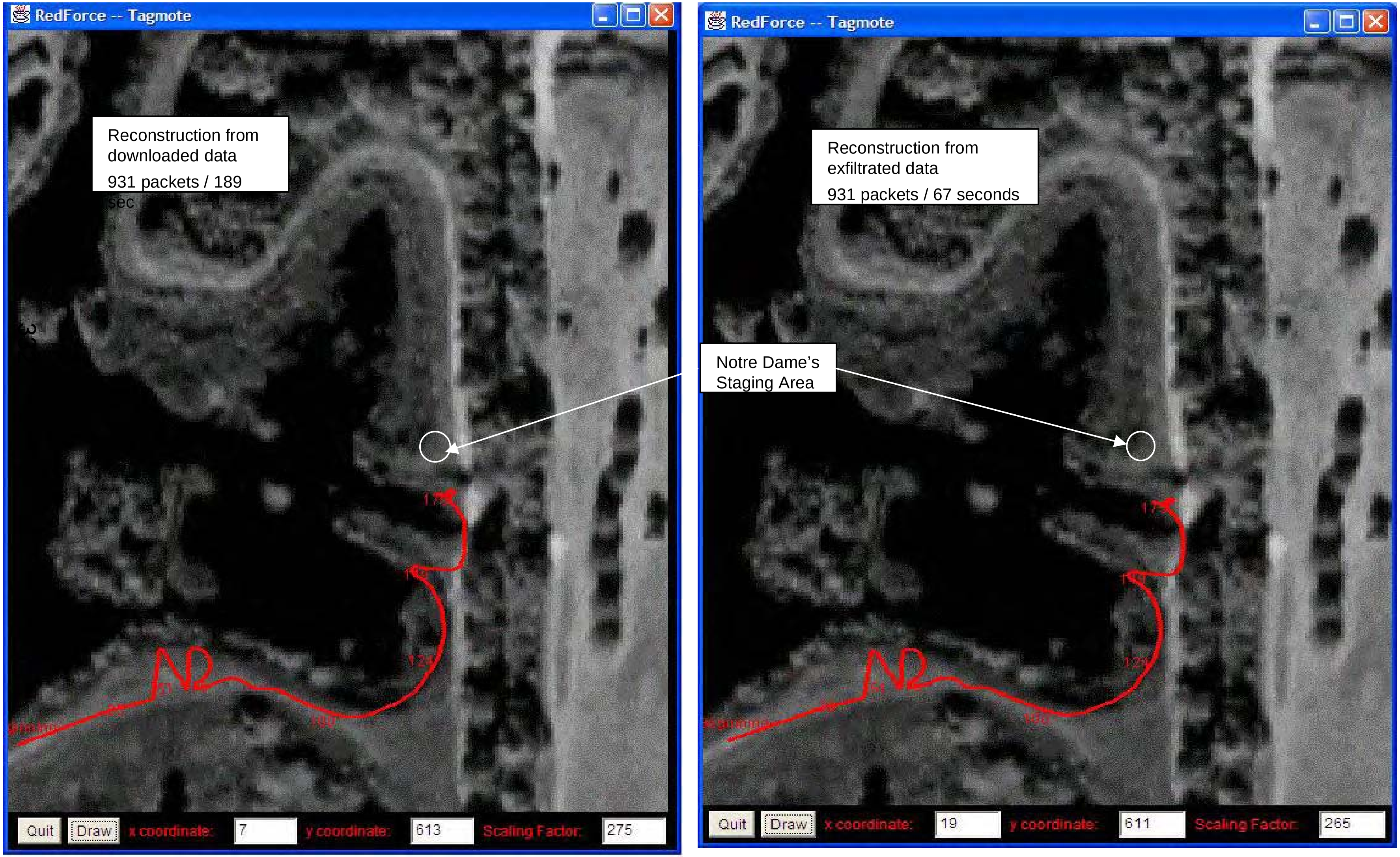
Reconstruction from downloaded data

851 packets / 189

Reconstruction from exfiltrated data

883 packets / 111

Experiment 3
(walk from UVA's site to ND's site)



RedForce -- Tagmote

Reconstruction from downloaded data

931 packets / 189 sec

x coordinate: 7

y coordinate: 613

Scaling Factor: 275

Notre Dame's Staging Area

RedForce -- Tagmote

Reconstruction from exfiltrated data

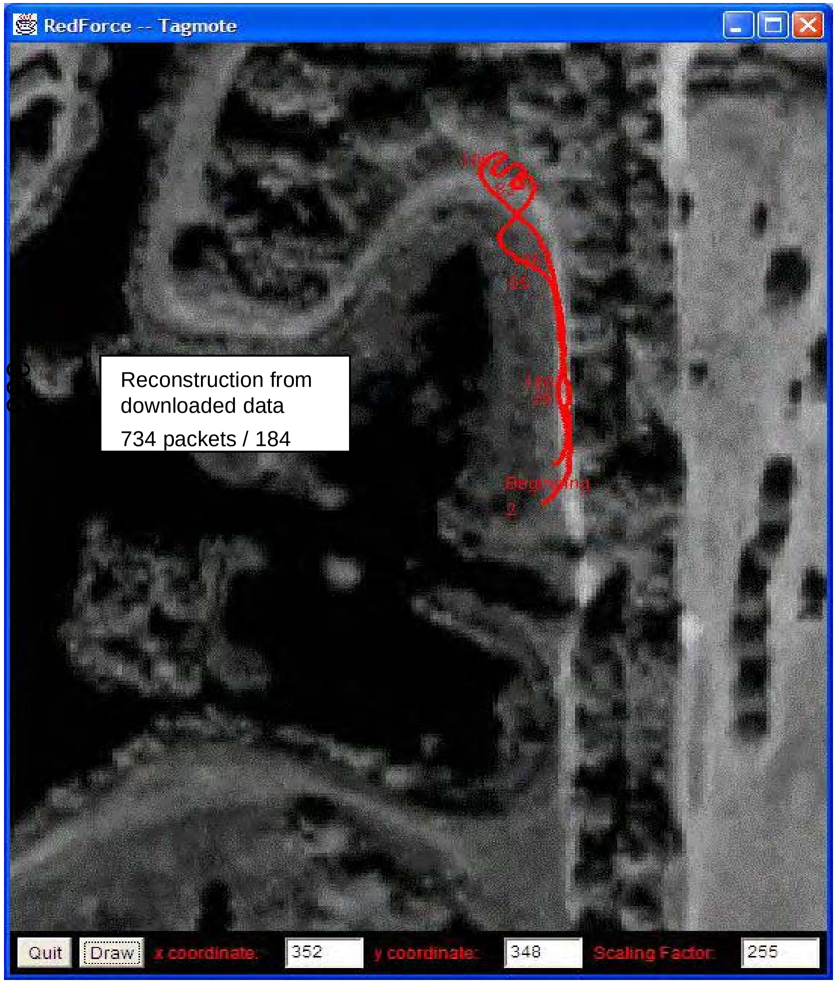
931 packets / 67 seconds

x coordinate: 19

y coordinate: 611

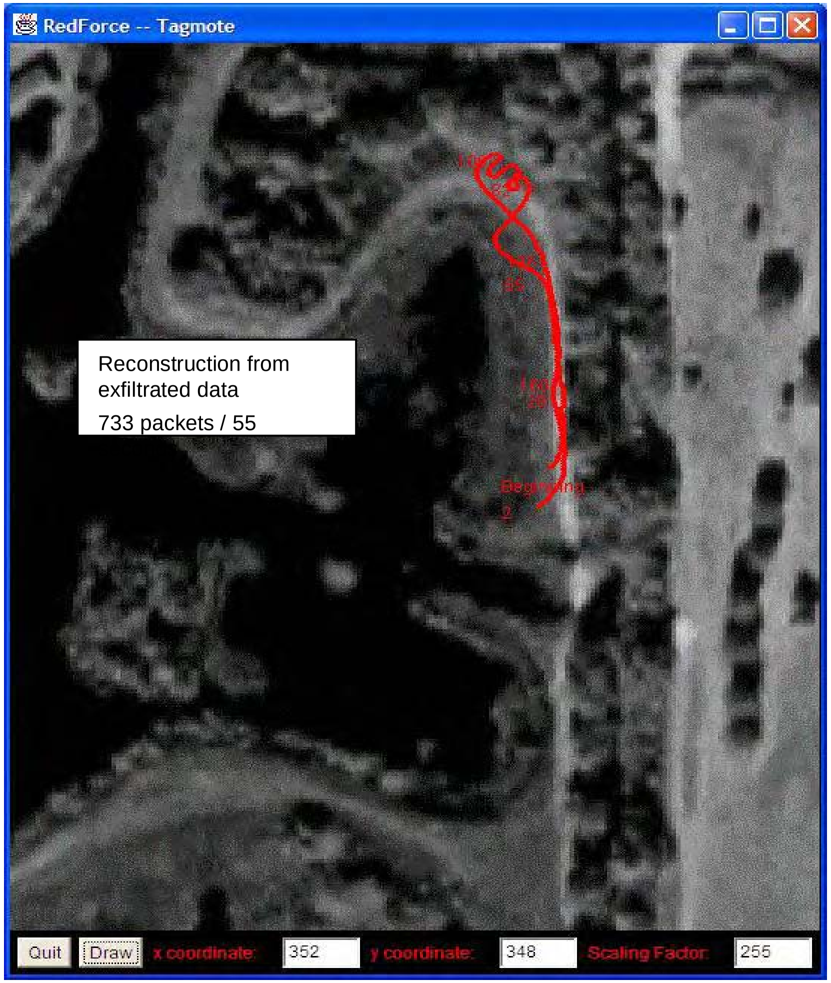
Scaling Factor: 265

Experiment 4
(walk north of ND’s site)



Reconstruction from downloaded data

734 packets / 184



Reconstruction from exfiltrated data

733 packets / 55

Caveats and Future Plans-MacDill AFB

Caveats	Future Plans
• Known starting point	• Addressed by using localization service
• Download from TagMote initiated by timer	• Enhanced with more intelligent software
• TagMote's size	• Determine minimum feasible size
• Performance issues – magnetic anomalies – sensor drift – subject variations	• Addressed by sensor fusion
• Download and exfiltration time	• Improvements in TagMote hardware and software, and in NEST middleware
• No subject's altitude capability	• Determined by adding barometer
• Currently limited to a person walking/running	• Enhanced by adding gyros and GPS

Real-time Configuration of Networked Embedded Systems

P.J. Antsaklis, Martin Haenggi, M.D. Lemmon
Dept. of Electrical Engineering
University of Notre Dame

DARPA/IXO NEST PI MEETING
June 2004, Charleston SC

Real-time Reconfiguration of Networked Embedded Systems

University of Notre Dame

Contract No. F30602-01-2-0526

AO No: L546/00

Period: 5/2001 – 5/2004. No Cost Extension to 12/2004

Panos J. Antsaklis (PI)

Department of Electrical Engineering

PI Phone No: 574-631-5792, Fax 574-631-4393,
antsaklis.1@nd.edu

DARPA Program Manager: Vijay Raghavan

Agent name & organization: William Koenig, AFRL/Rome

Subcontractors and Collaborators

Senior Personnel
Panos J. Antsaklis, Michael D. Lemmon, Martin Haenggi
Electrical Engineering, University of Notre Dame

Post-doc
Marian Iordache

Graduate Students
Luis Montestruque, Lei Fang, Hui Fang, Ioannis Koutroulis,
Min Xie, Nicholas Kottenstette

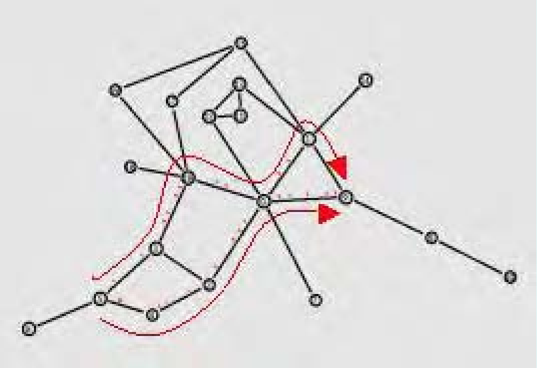
OSU-Extreme Scaling. Attended April meeting in Columbus.

Univ. of Notre Dame

Real-time Reconfiguration of Networked Embedded Systems

Problem and Challenge

Real-time reconfiguration of middleware services in networks of embedded systems



Real-time+ Distributed Algorithms + Feedback Loops+Probabilistic Analysis+Formal Verification

New Ideas

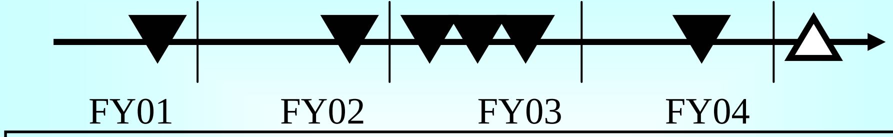
- Model-based clock synchronization and security for heterogeneous networks of embedded systems
- Adaptive reconfiguration for robust scalable middleware services in response to abrupt network changes
- Real-time network management through probabilistic analysis and formal verification.
- Exploiting feedback in distributed algorithms
- Quasi-regular topologies in Large Networks

Impact

- Middleware for ad hoc networks of embedded systems.
- Real-time reconfiguration of streaming connections in ad hoc wireless networks.

- Red Force Tagging
- Large Network Architectures

Schedule



3QFY01: Project Start
3QFY02: Berkeley OEP Demo
1QFY03: Demo of Model Problem (robot formation control)
2QFY03: Delivery of TinyOS Service Components
3QFY03: Integration with Group I (Red Force Tagging)
3QFY04: Red Force Tagging & Networking Enhancements
1QFY05: Enhancements & Technical Transfer to Military

Problem Description and Objectives

Adaptive reconfiguration in fine grained networks of embedded systems that interact tightly with the physical world so to maintain Quality of Service (QoS.)

Project Tasks:

- Clock-synchronization:
  Develop clock-sync algorithms for ad hoc networks.

- Adaptive Reconfiguration
  Fast reconfiguration of NEST services
  in the presence of abrupt network changes

- Red Force Tagging for SOCOM

Project Status

PROGRESS SINCE LAST MTG.

- Red Force Tagging
  - TagMote hardware enhancements
  - TagMote protocol and algorithm enhancements
  - Download and network service throughput improvements

- Random vs. Quasi-Regular Sensor Networks
  - Analysis of inter-node distances in random networks
  - Equivalence of two different quasi-regular networks

Goals and Success Criteria

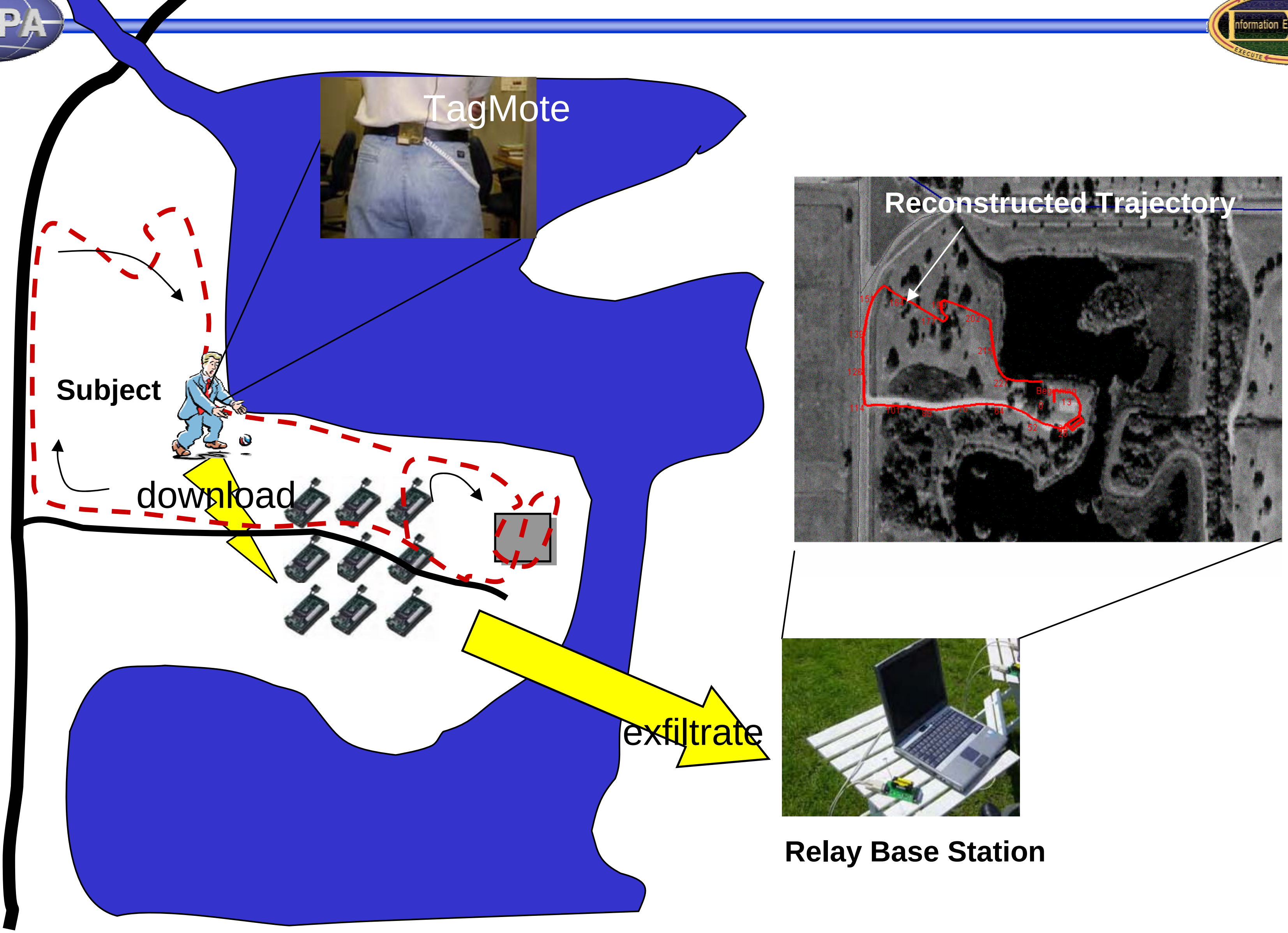
- **Red Force Tagging**
  - Hardware enhancement-design and implementation of microprocessor board
  - Download and exflitration service throughput enhancements

- **Large Networks of Embedded Sensors**
  - Identify efficient architectures that provide high throughput and reliability at low power consumption: Quasi-regular networks

Outline

- Red Force Tagging
  - Project Description
  - Enhancements
- Large Networks of Embedded Sensors
- Program Issues

Red Force Tagging

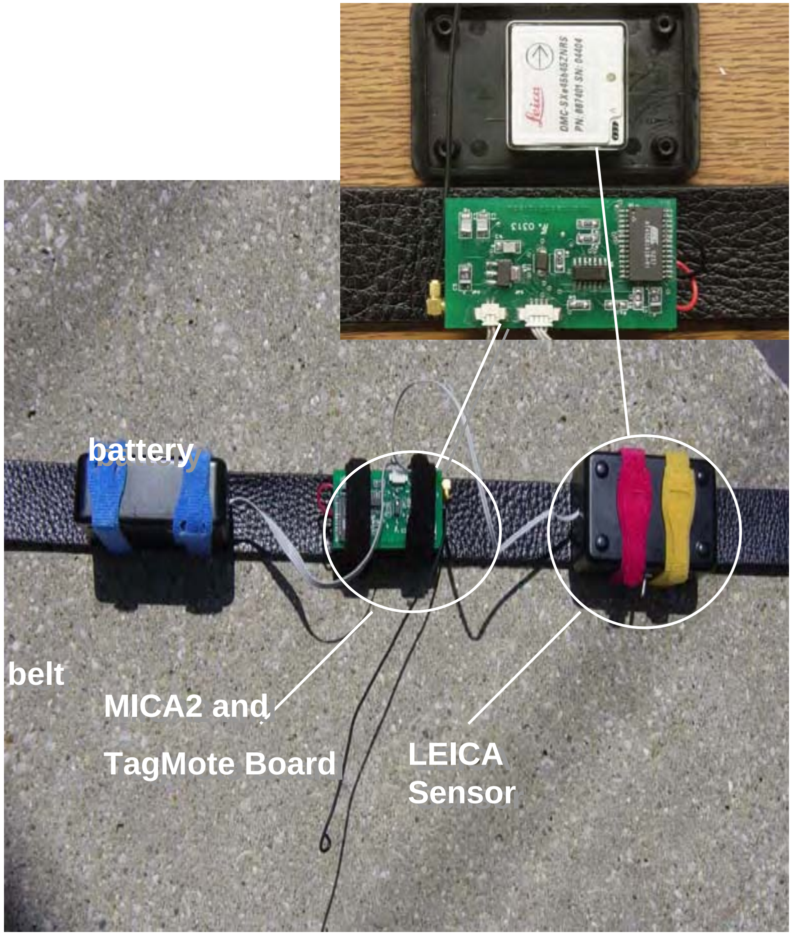


Benefits

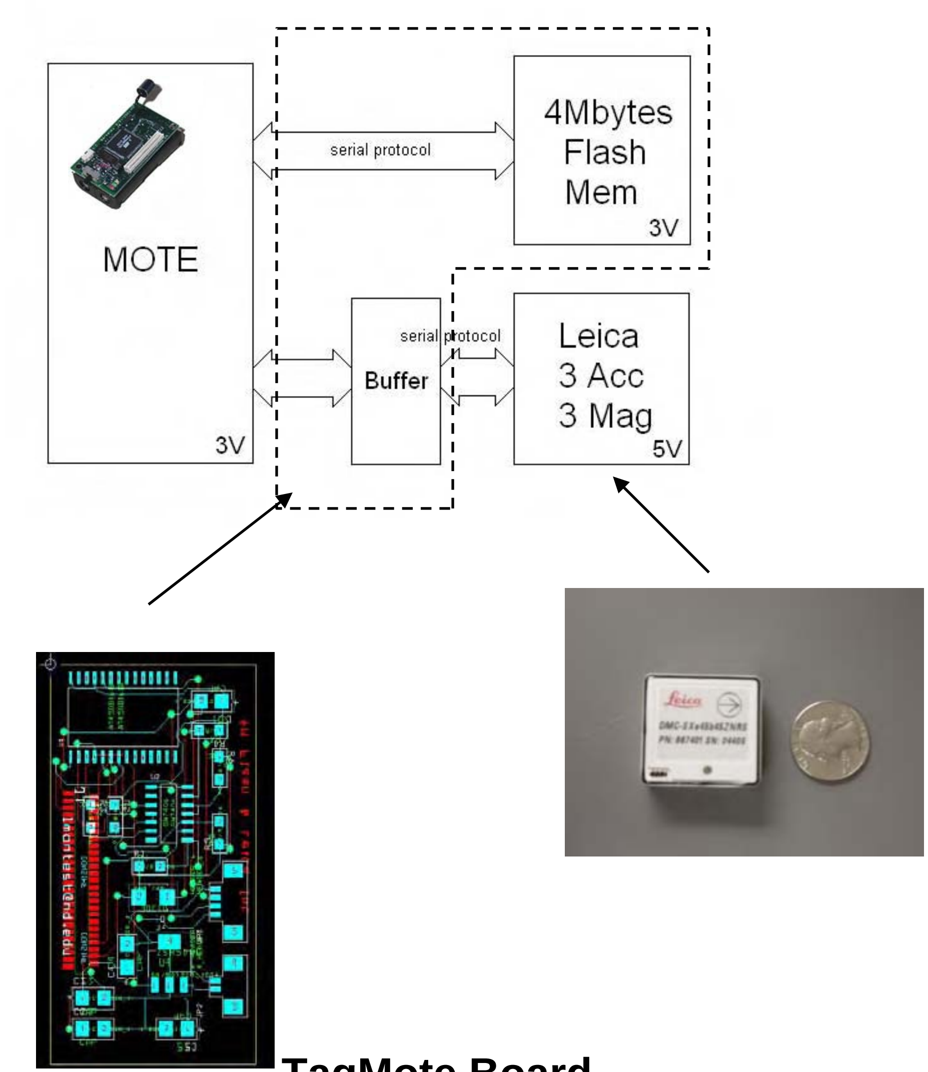
- Discovering Red Force in GPS-denied areas
- Does not rely on pre-deployed wired network
- Can be used by Blue Force
- Multi-sensor data collecting platform
  - Extensible to other sensor types : biochem, radiation

Technical Approach – TagMote

- **Designed and Built a Special “TagMote”**
  - **TagMote board**
    4 MByte Flash Memory
  - **Leica Sensor**
    3D-accelerometer
    magnetic compass
  - **MICA2 for communication**
  - **Attached to a belt**



TagMote Hardware Overview

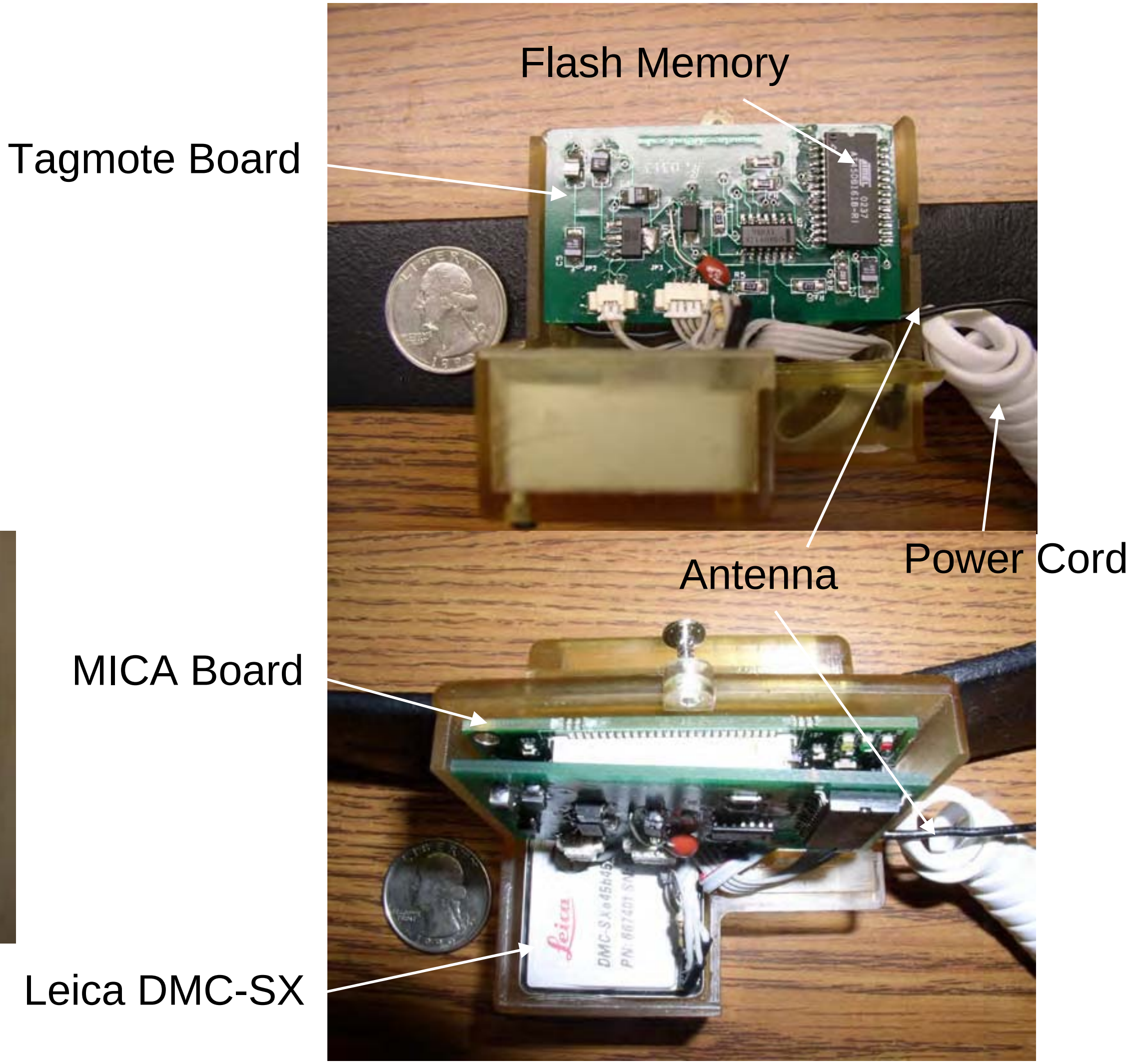


TagMote Board

Leica Module

TagMote: Design II

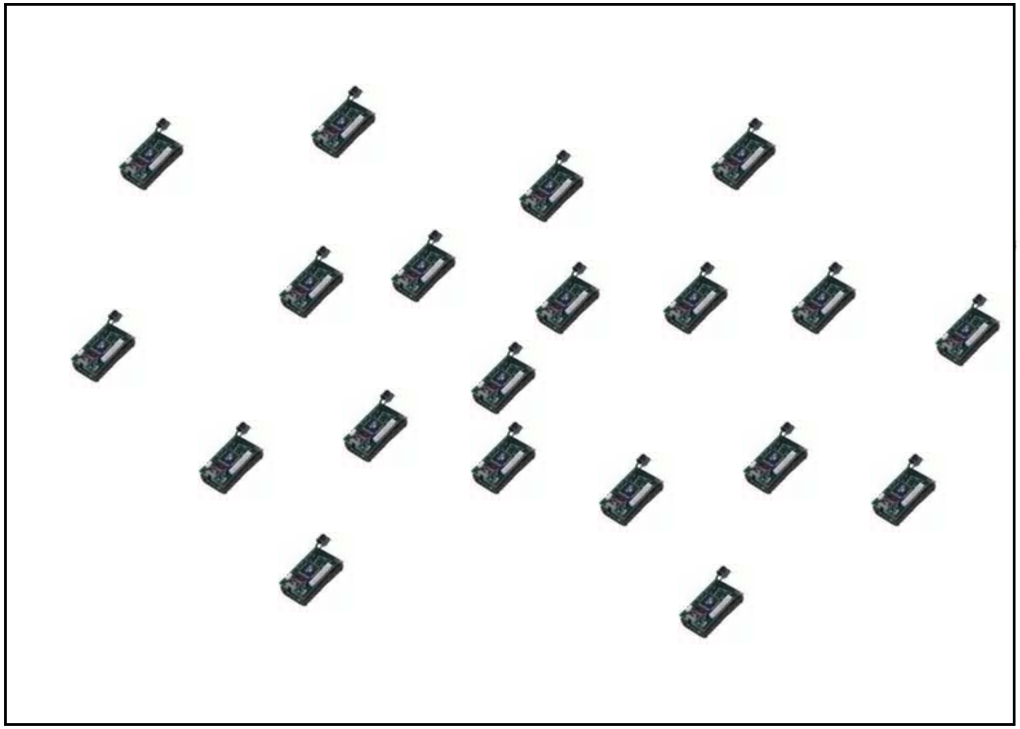
- Compact: 65 ×58×24 mm
- Lightweight:100 grams



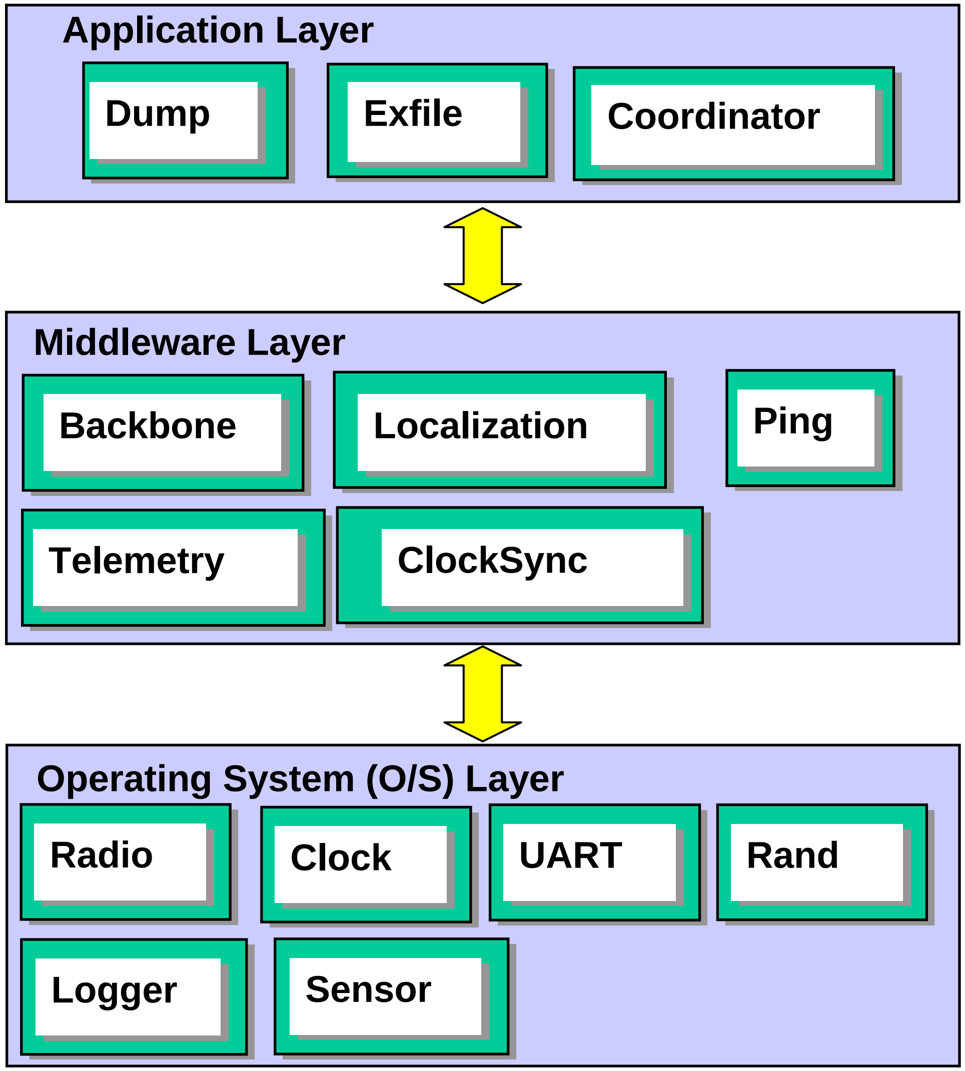
Technical Approach – NEST Middleware

- NEST Middleware Challenges
  - Download large amounts of data from TagMote to NEST
  - Exfiltrate large amounts of NEST data to SOCOM relay
  - Reliable and Timely Data Delivery
- Solutions
  - Multi-hop Streaming service (link reservation)
  - Explicit packet acknowledgement
  - Route reconfiguration (based on link level timeouts)

Clear Basic Service Cell



Red Force Tagging Middleware Architecture



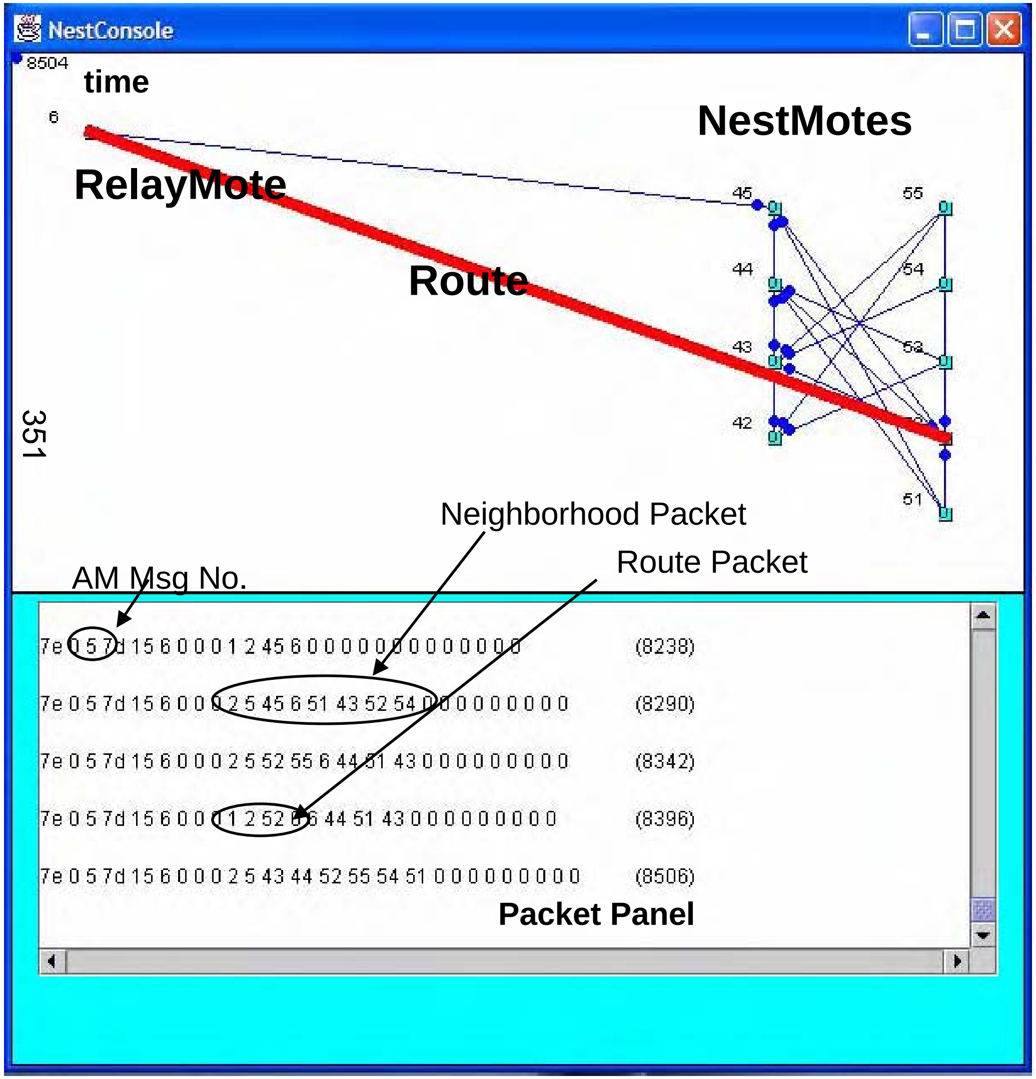
TagMote Software Tasks

- **Tagmote software implements:**
- Compensation for adverse sensor orientation, calibration, filtering;
- Collection of data from sensors and storage on a non-volatile memory;
- Retrieval of data from memory, processing and transmission;
- Detection of NEST and data download;
- Pre-processing of raw sensor data before data download.

- **NEST network services provide**
- Synchronization: establishes global time reference;
- Backbone generation: establishes min-hop route to RelayMote;
- Streaming service: streams large blocks of data from NestMote to RelayMote;
- Download service: streams large blocks of data from TagMote to NestMotes.

- **Trajectory Post-Processing implements**
- Java programs on PC connected to RelayMote;
- Reassembly of data packet;
- Compensation for missing data;
- Reconstruction of complete trajectory;
- Display of trajectory over area map.

Telemetry Service



```
7e 0 5 7d 15 6 0 0 0 1 2 45 6 0 0 0 0 0 0 0 0 0 0 0 0 0        (8238)
7e 0 5 7d 15 6 0 0 0 2 5 45 6 51 43 52 54 0 0 0 0 0 0 0 0 0     (8290)
7e 0 5 7d 15 6 0 0 0 2 5 52 55 6 44 51 43 0 0 0 0 0 0 0 0 0     (8342)
7e 0 5 7d 15 6 0 0 0 1 2 52 0 6 44 51 43 0 0 0 0 0 0 0 0 0      (8396)
7e 0 5 7d 15 6 0 0 0 2 5 43 44 52 55 54 51 0 0 0 0 0 0 0 0 0    (8506)
```

351

- After backbone built, TLM service started
- AM msg 5
  - Neighborhood packets
  - Route packets
- RelayMote uses TLM service to build a graphical view of the network
- Displays position, neighborhoods, and routes (Mote SerialID’s encode location)
- Dumps the decoded data packets to a file

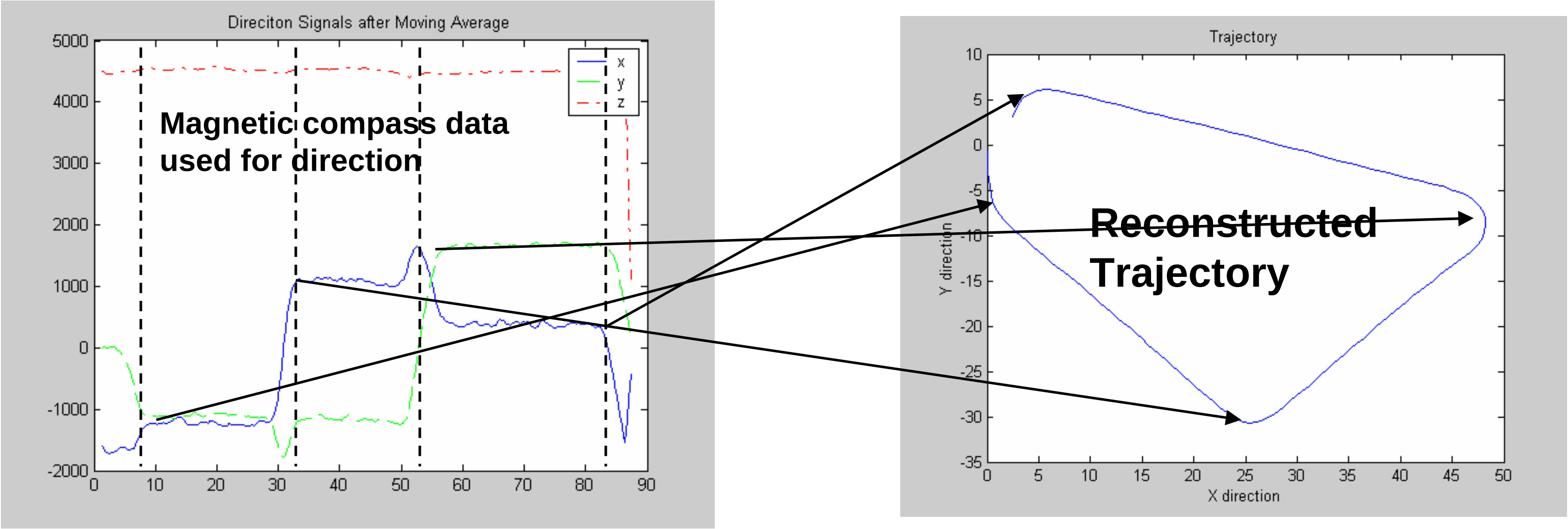
Technical Approach – Algorithms
Dead Reckoning

Data Generated by TagMote

- **Step length estimation**

$$K \times \sqrt[4]{a_{z\max} - a_{z\min}}$$

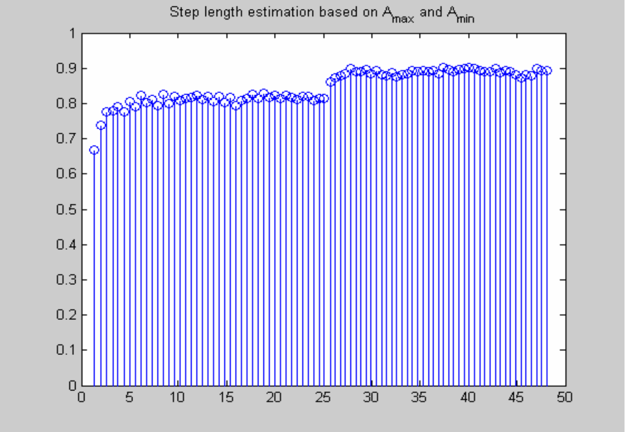
- Data from accelerometer and magnetic compass
- Trajectory reconstructed from step count, step length, and direction
- Part of processing done on TagMote Bulk of trajectory reconstruction done at RelayMote

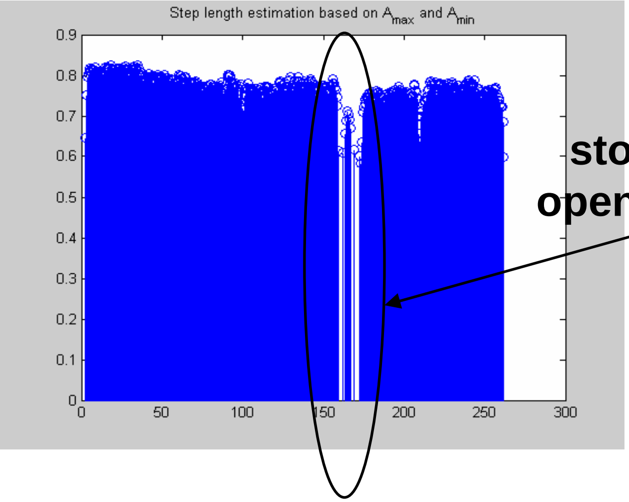


Trajectory Reconstructed at RelayMote

Gait Analysis

Change in speed & stride length

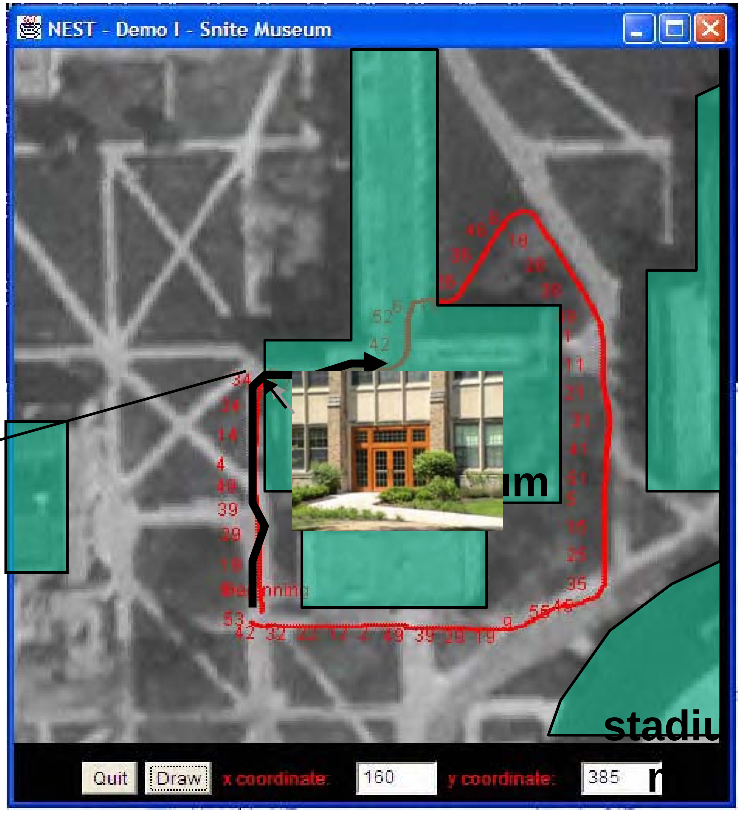




stop to open door

Acceleration and step length histories can be used to analyze subject's gait

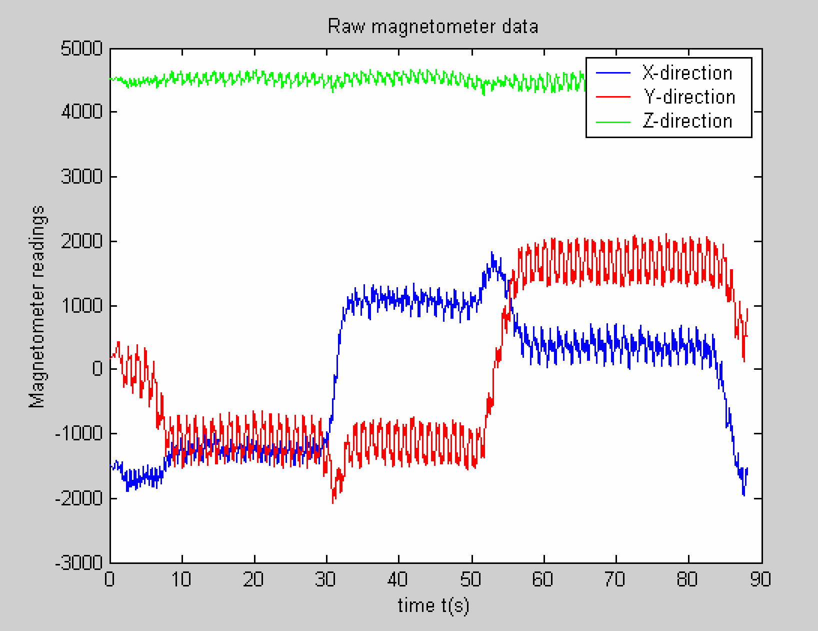
- gait changes walking/running
- stopping to open doors



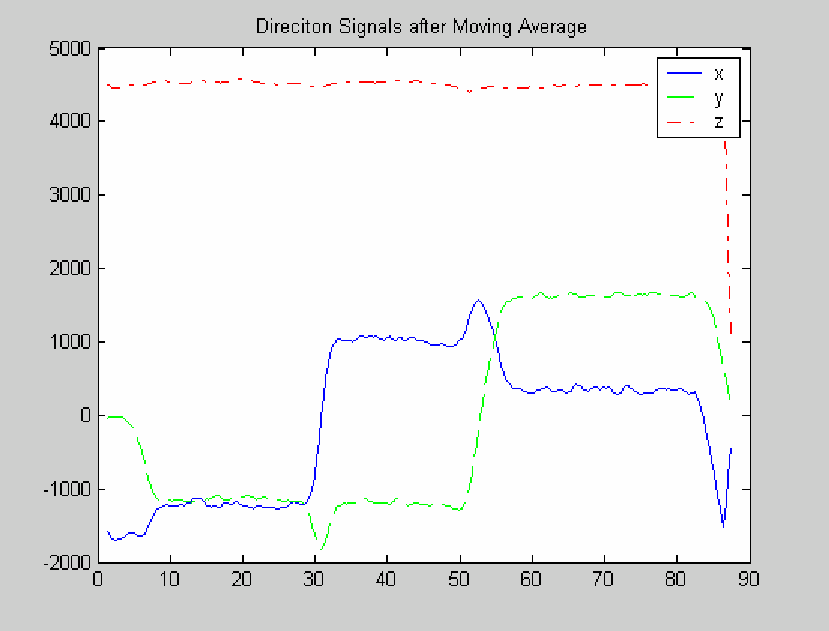
Heading Estimation

- Moving-average

Raw Compass Data

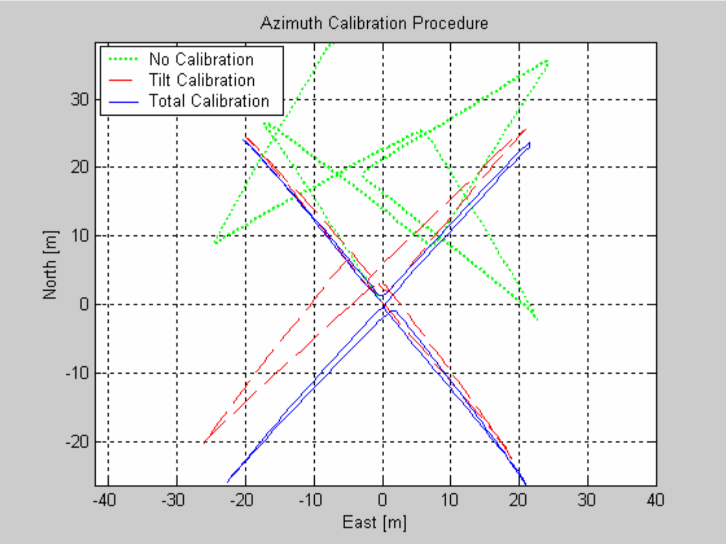


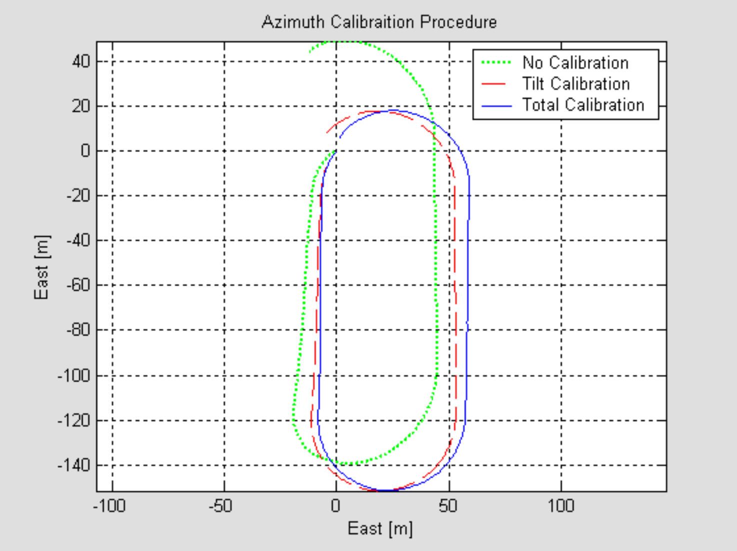
After Moving-average



- Dynamic calibration: Tilt compensation using 3-D accelerometer data
- Static calibration: Magnetic field distortion (e.g., due to human body)

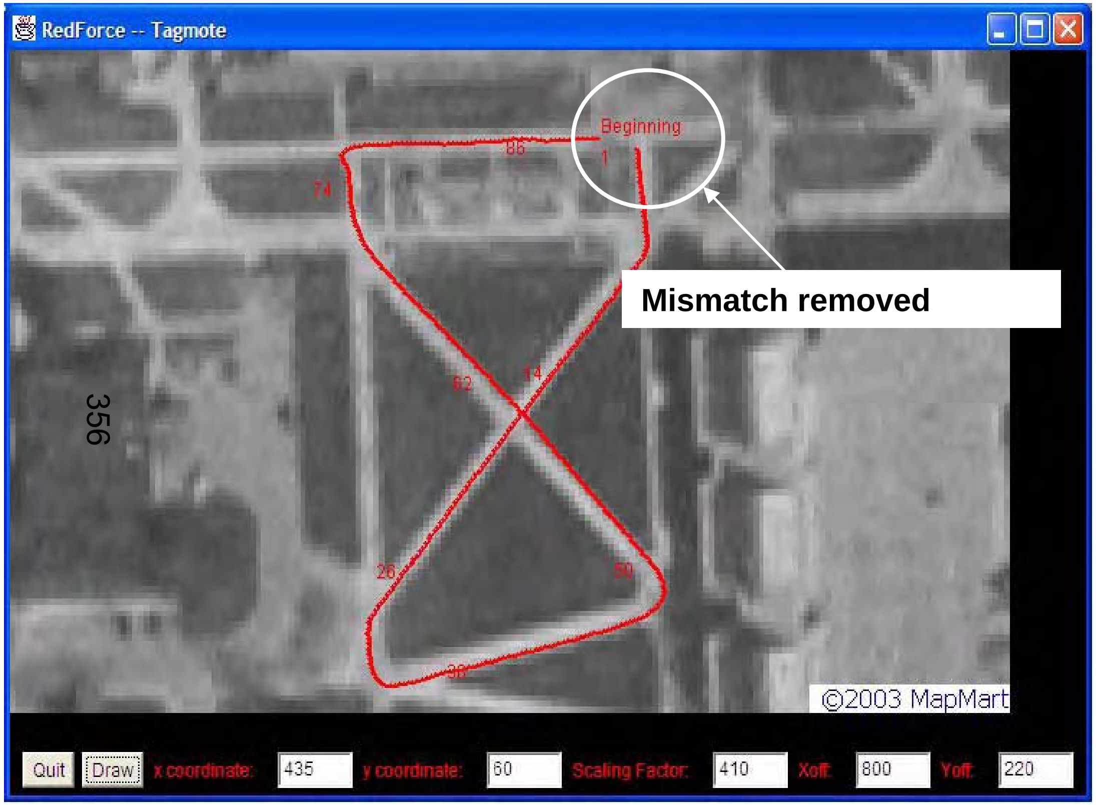
Heading Calibration





- Calibration is essential in order to obtain good performance
- NEST middleware provides a method for in situ calibration of MICA sensors.
- Experimental heading error: 6.8° without calibration, 1.1° after dynamic and static calibration
- NEST localization services can be used to provide “absolute” position fixes that can be transmitted to TagMote.

Post Reconstruction Analysis



RedForce -- Tagmote

Mismatch removed

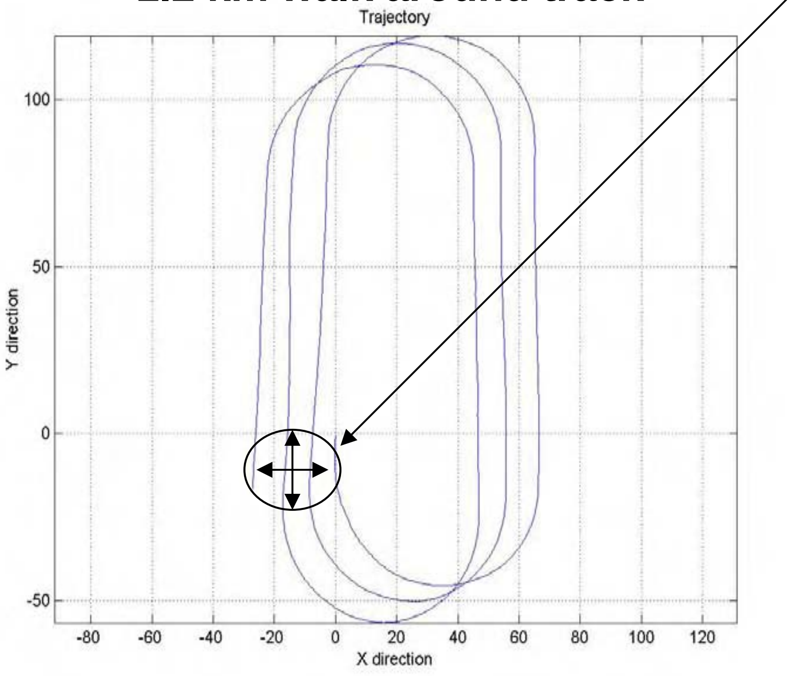
©2003 MapMart

x coordinate: 435 y coordinate: 60 Scaling Factor: 410 Xoff: 800 Yoff: 220

- Calibration is essential to obtain good performance
  - Dynamic (Tilt) Calibration
  - Static Calibration
- Calibration constants, however, are never exact.
- Post Reconstruction Analysis fits exfiled data to known waypoints.

Overall Performance

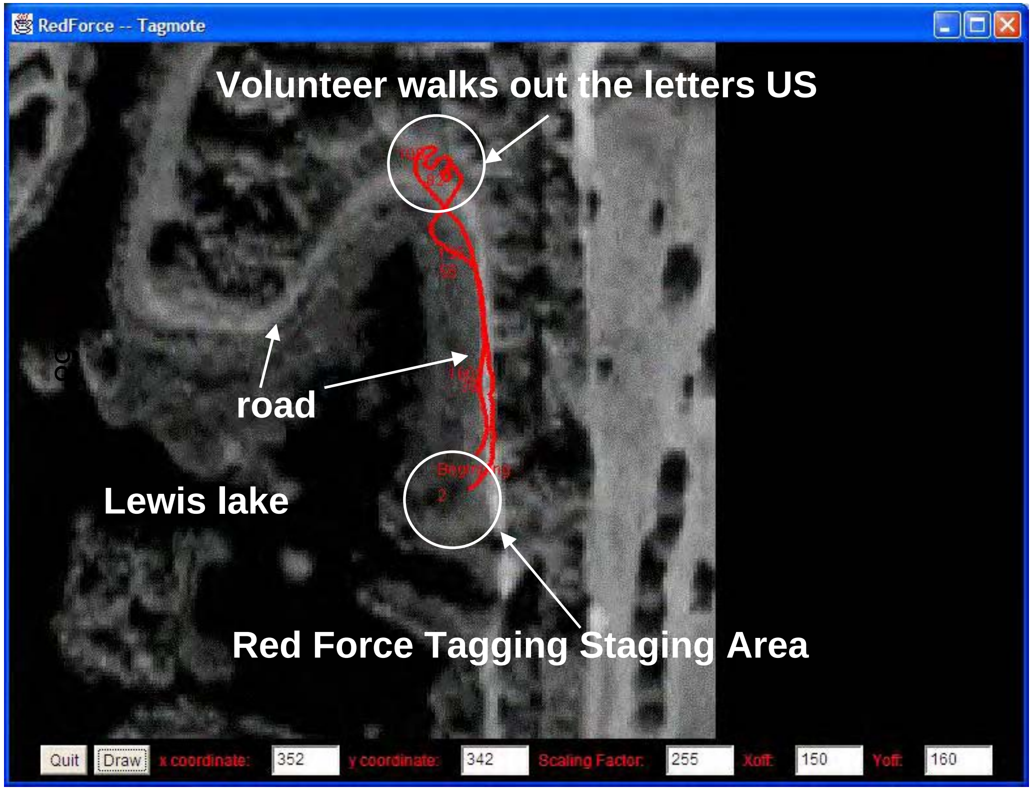
1.2 km walk around track



- Distance Accuracy: ±3% for same subject
- Heading Accuracy: ±1°
  - Example: (27m,16m) error over 1200-meter walk

- Download time
  - 3 min. walk = 3 min. download
  - 70 bytes/sec (data)
- Exfiltration time
  - 3 min. walk = 1 min. exfiltrate
  - 200 bytes/sec (data)
- Reliability: 90-100 %

Performance-Field Experiment



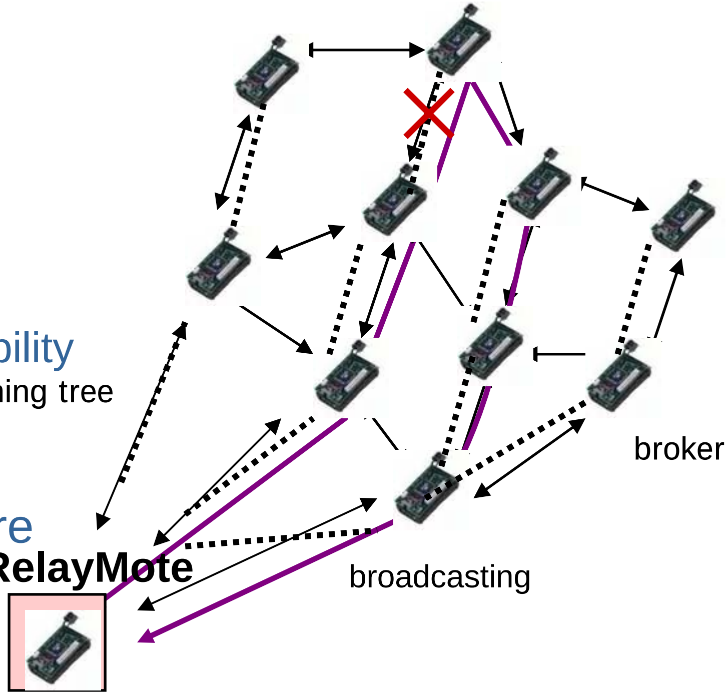
Download trajectory data to NEST, exfiltrate the data to the Relay, reconstruct and display the subject’s trajectory.

Outcome:

- 100 percent packet recovery
- 20 meter accuracy over 1 km run
- 3 minute download time/ 3 minute walk (70 data bytes/sec)
- 1 minute exfiltration time/ 3 minute walk (200 data bytes/sec)

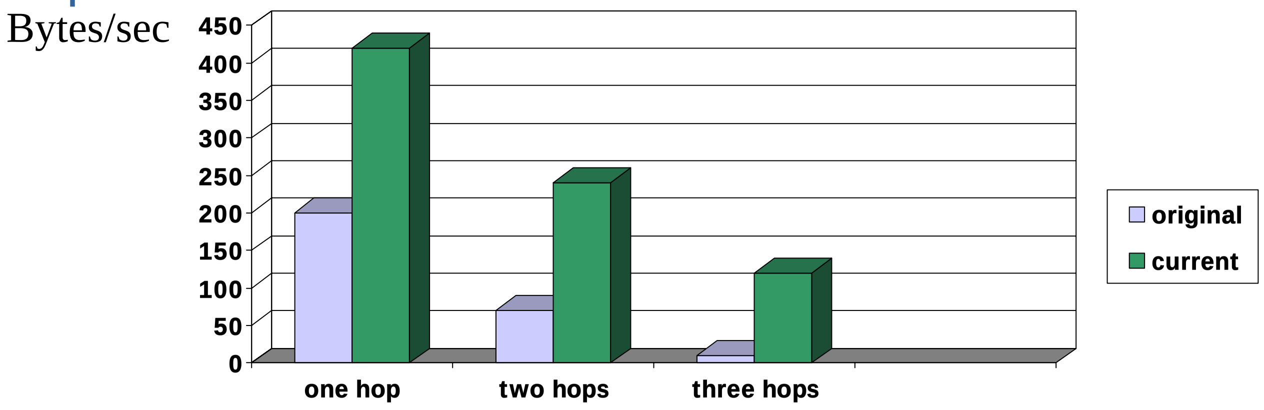
Enhanced Backbone Formation Service

- Red Force Tagging Performance (field experiment)
  - – Limited Connectivity in Spanning Tree
    - only kept a single next-hop in the tree
- Enhanced Backbone service
  - – Keeps track of "retries" as estimate of link reliability
  - – Keeps track of multiple "next hops" in a mote's neighborhood
- Provided a highly connected routing structure upon which to exfiltrate TagMote data



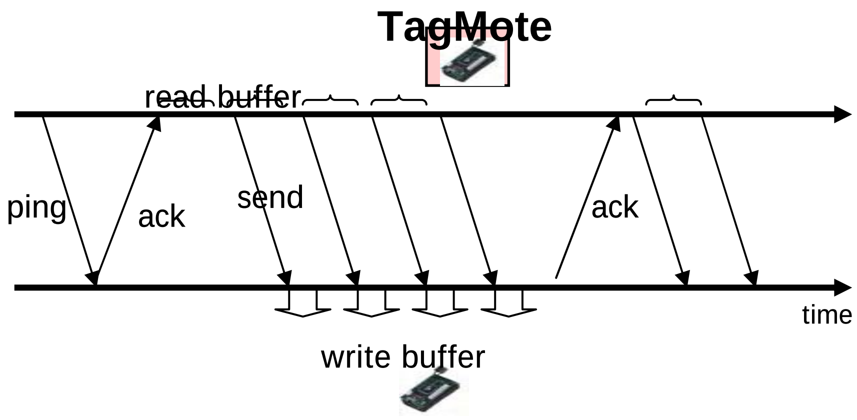
Exfiltration Service

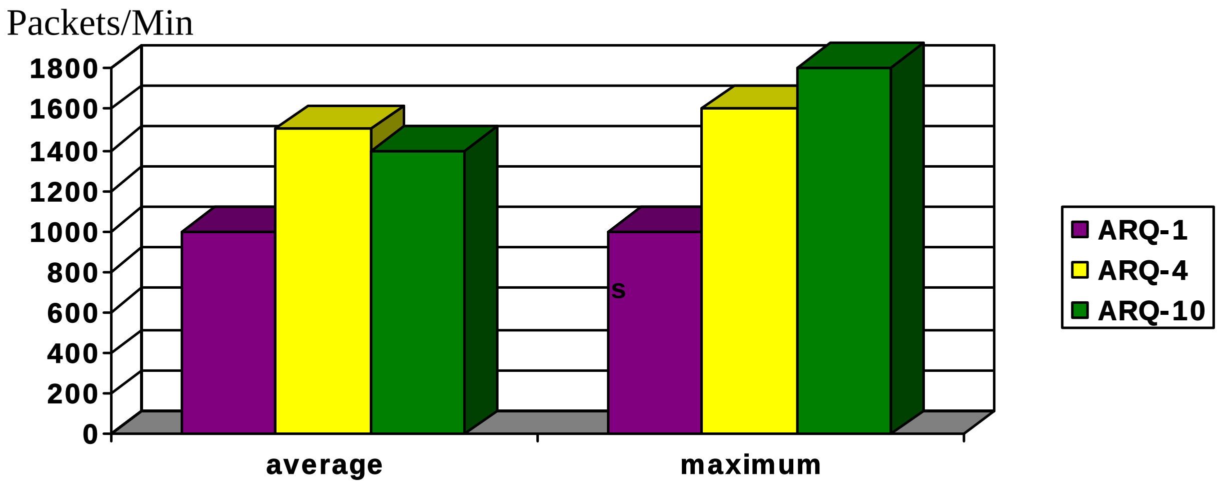
- (Earlier) Performance
  - Low throughput caused by limited connectivity in spanning tree
  - Deadlocks due to missed messages
- Enhanced Exfiltration Service
  - Switches next-hop based on throughput estimate
  - Watchdog timer to reset “dead” motes in a basic service cell
- Performance of Enhanced Service
  - linear scaling of throughput with number hops
  - higher throughput
  - multiple tests on a 50-node NEST field



Enhanced Download Service

- Red Force Tagging (Florida) Performance
  - Slow download rates
- Enhanced Download Service
  - ARQ-N scheme
- Enhanced Performance
  - field tested
  - improved throughput

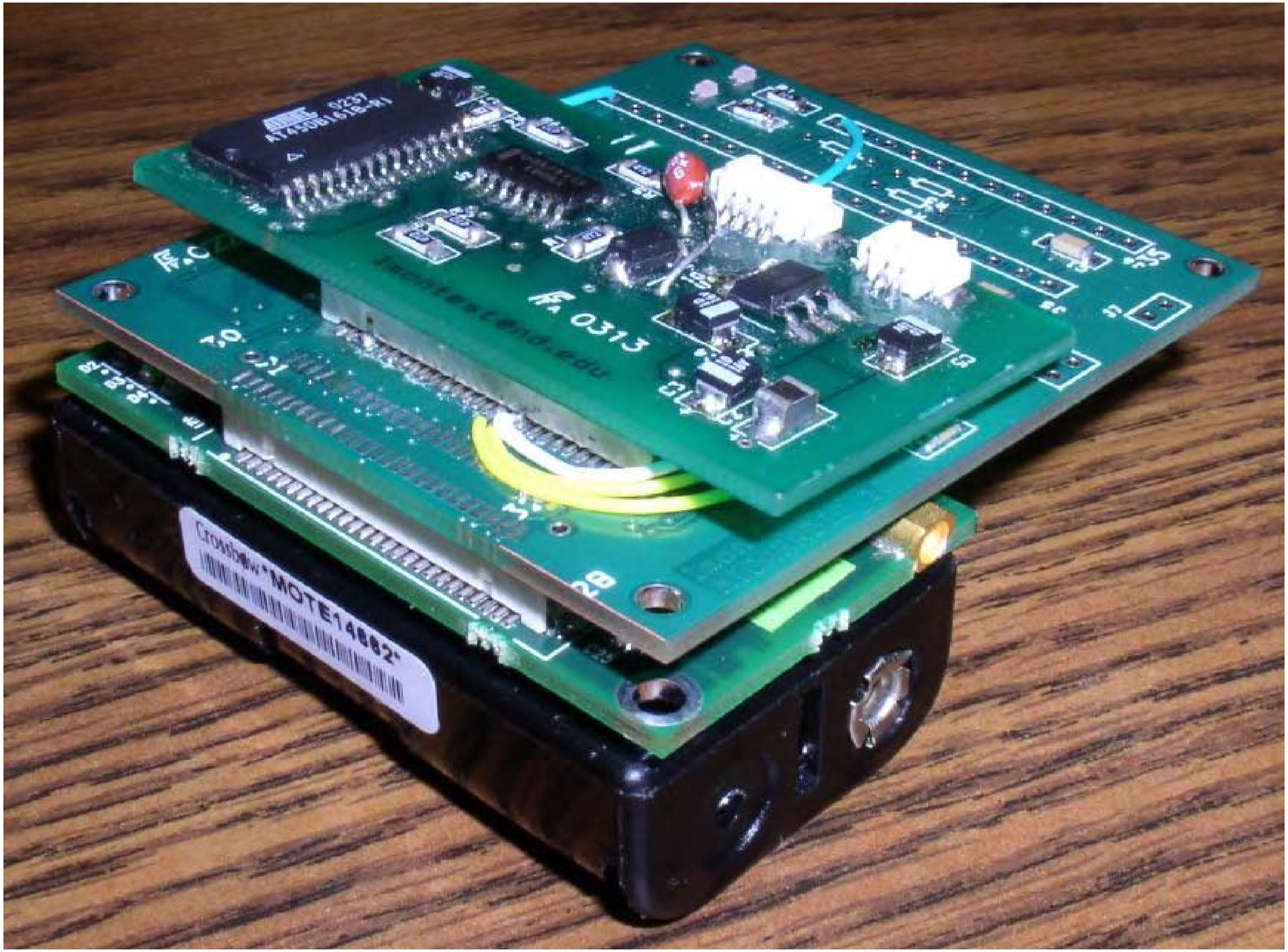




TagMote HW Redesign

- Increased speed of download
- Increased flexibilty
  - Easier to add additional sensors
  - Easier to introduce more efficient algorithms, add new processing functions
- Keep same processor type for easy interface

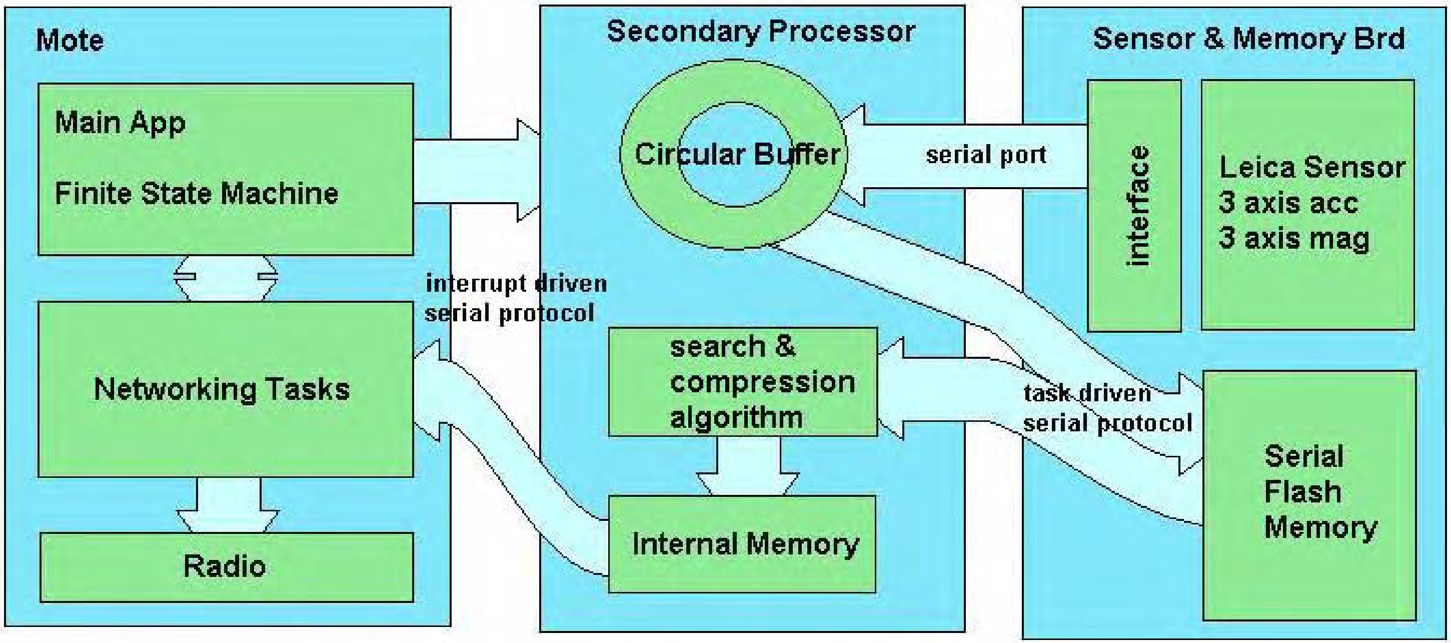
TagMote HW



TagMote New HW

- Original TagMote Hardware:
  - Mica 2
  - Sensor and Flash Memory Board
- New TagMote Hardware incorporates additional processor:
  - 8 MHz ATMEGA 128L
  - Standard Interface to Mica2 programmer
- Additional Processor (Slave) communicates with Mica2 (Master) using custom synchronous serial interface.
- Slave controls Sensor and Flash Memory card.
- Master in charge of Networking tasks

New HW Design-Adding Microprocessor



TagMote New HW

Master Tasks:

- Coordinate overall behavior.
- Find Network (ping).
- Identify node to download to.
- Forwarding of data to network (dump).

Slave Tasks:

Configure Leica Sensor.

Acquire data from Sensor.

Process data (min max search).

Compress data.

Storing data in Flash.

Forwarding data from Flash to Master when needed.

Differential Storage and Transmission

- In the previous versions, we used 16 bytes of data:
  2 bytes for each of Gx, Gy, Gz, Ax, Ay, and Az

- In the new differential version we use 24 bytes:

Bytes 0 ... 3 contain: Time of data set 1 (28 bits),

data type: min or max (1 bit), packet type (compressed or not, 1 bit), differential time of data set 2 (2 bits).

The 32 bit word of bytes 0 ... 3 is unique, also used as msg ID

Bytes 4...23 contain: the differential measurements.

Tests and Experiments

Tagmote Design Innovations	Download (in % of walking time*)	Exfiltration (One-hop Case)
One Processor	106%	31%
Preproc. on master after flash storage	156%	31%
Preproc. on slave after flash storage	89%	31%
Preproc. on slave before flash storage	44%	31%
As before with differential compression	28%	17%

*The total walking time with 4MB Flash is up to 16 hours.

Additional TagMote HW Considerations

- Redesign the TagMote without the Leica sensor to reduce size and costs

- Programmable Logic
  - future, reduction of production costs

Outline

- Red Force Tagging
- Random vs. Quasi-Regular Sensor Networks
- Program Issues

Distances in Random Networks

Consider a Poisson point process of intensity λ in m dimensions.

The distance R_n to the n-th neighbor in an m-dimensional random network has the pdf (generalized Weibull distribution):

$$f_{R_n}(r) = e^{-\lambda c_m r^m} \frac{m\,(\lambda c_m r^m)^n}{r(n-1)!},$$

Where $c_m r^m$ is the volume of the m-sphere. In particular, $(R_1)^m$ is exponentially distributed.

Energy Consumption in Random Networks

With a path loss exponent α, the expected energy consumption is proportional to

$$\mathbb{E}[R_n^\alpha] = \left(\frac{1}{\lambda c_{\phi,m}}\right)^{\frac{\alpha}{m}} \frac{\Gamma(n+\frac{\alpha}{m})}{\Gamma(n)},$$

so α and m play complementary roles. Indeed, for most statistics, only the ratio of the path loss exponent and the number of dimensions is relevant.

Energy Consumption in Random Networks

Maximum energy consumption in an M-hop hearest-neighbor route:

Denote the hop distances by $R_1, \ldots, R_M$.
Since R^m is exponentially distributed, we know from selection combining

$$\mathbb{E}[\max\{R_1, \ldots, R_M\}^m] = \mathbb{E}[R^m] \sum_{k=1}^{M} \frac{1}{k}$$
$$\gtrapprox \mathbb{E}[R^m] \left(\ln(M) + \gamma_{\text{em}}\right) ,$$

and, with Jensen's inequality (for $\alpha \geqslant m$),

$$\mathbb{E}[\max\{R_1, \ldots, R_M\}^\alpha] \geqslant \left(\mathbb{E}[\max\{R_1, \ldots, R_M\}^m]\right)^{\alpha/m}$$
$$\gtrapprox \left(\mathbb{E}[R^m](\ln M + \gamma_{\text{em}})\right)^{\alpha/m} .$$

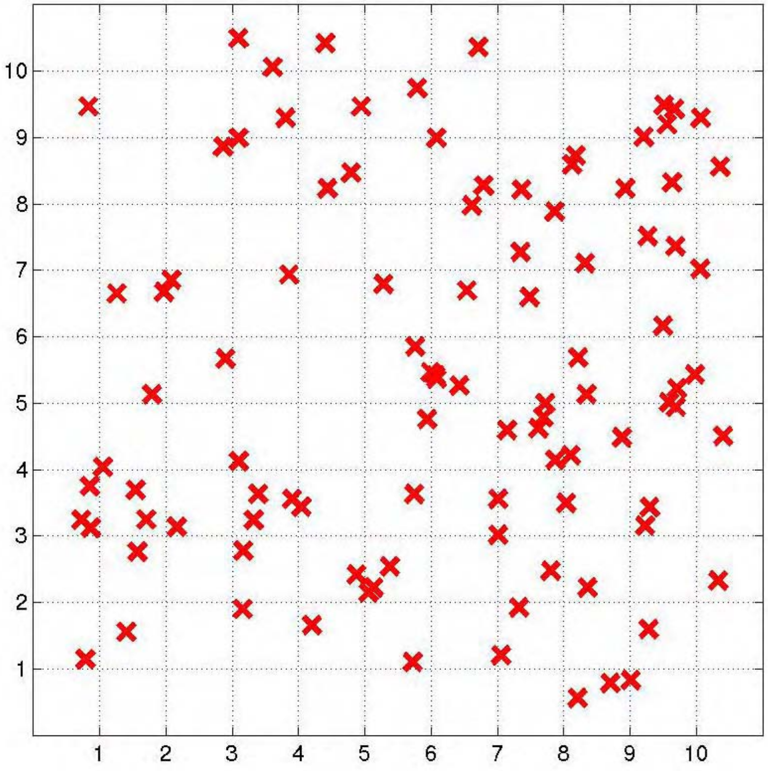
So the expected maximum energy consumption grows with $(\ln M)^{\alpha/m}$.

Comparison of Regular and Random Networks

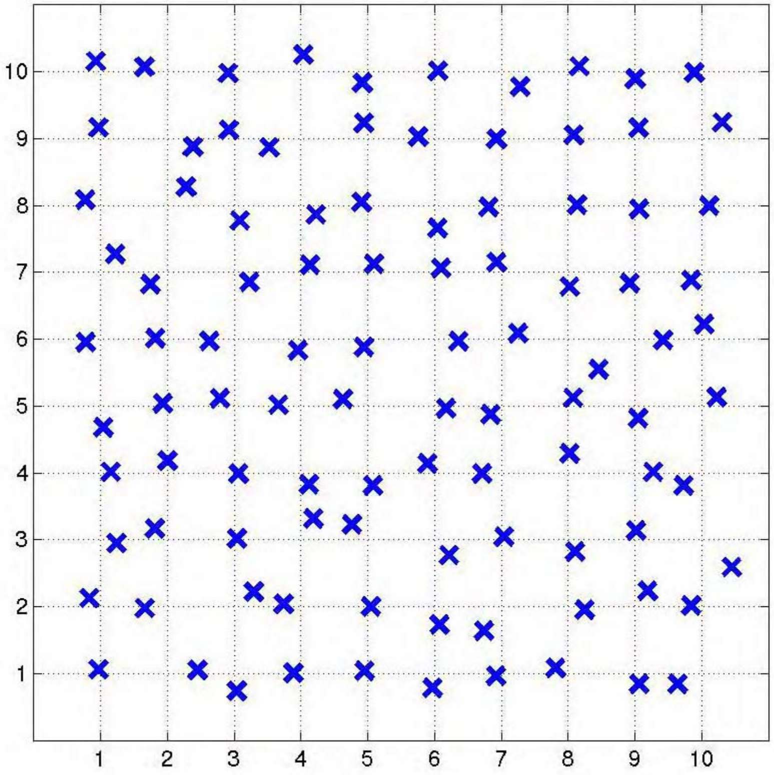
- Random networks are difficult to deal with.
- Perfectly regular topologies are ideal (for sensing and communication) but hard to achieve. If nodes are placed at *approximately* equal distances, we speak of a quasi-regular network.
- The deviation from the ideal grid points can be modeled by a Gaussian distribution. The variance controls the regularity of the grid.

Random and Quasi-Regular Networks

Two networks with 100 nodes:



Random distribution



Quasi-regular distribution with
σ=0.2 on integer grid.
“Gaussian network”

Quasi-Regular Networks

Still, for this Gaussian network, nodes need to be placed manually. What if we drop more nodes randomly and pick for every grid point the closest node as an active node? Denote the network that results as a “**partial random network**”.

The active subset of nodes also form a quasi-regular network.

Theorem:

The node distribution in a Gaussian network with variance σ^2 and a partial random network of intensity λ are equivalent if $\lambda=1/(2\pi\sigma^2) \gg 1$.

For example: For $\sigma=0.2$, $\lambda=4$. I.e., we have to put 4 times more nodes.

Quasi-Regular Networks II

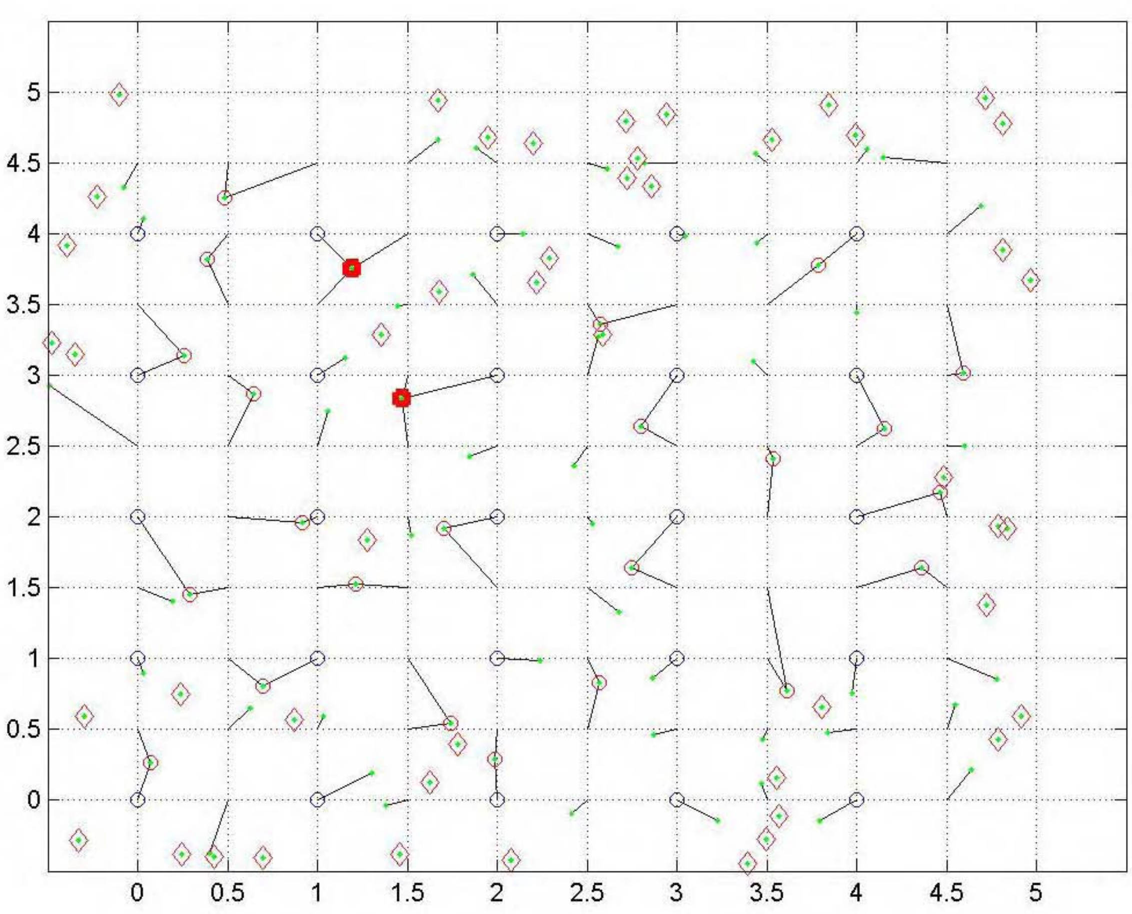
- So, rather than manually placing nodes, we can deploy them in larger numbers and use only a subset.

- For our example (λ=4), only ¼ of the nodes are active at a given time. The rest can be put to sleep.
  To balance energy consumption, it makes sense to shift the integer grid by ½ after a certain period and again choose the closest nodes to the new grid points to be active.
  The problem is that some nodes may need to be active in more than one period.

Quasi-Regular Networks

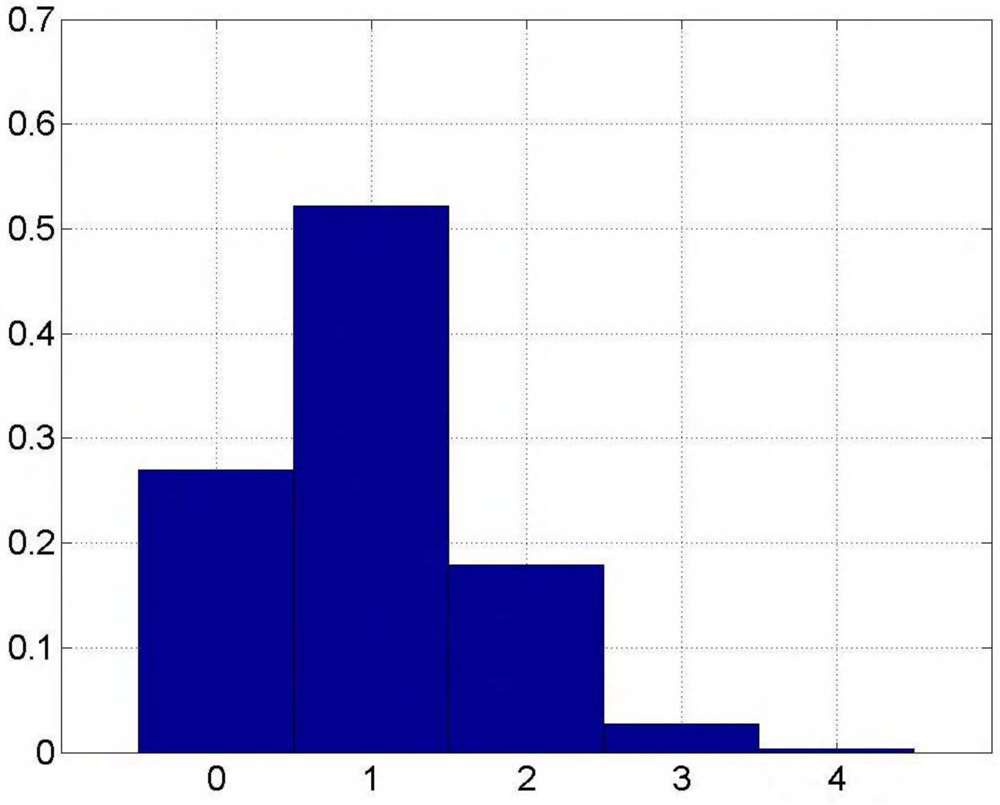
With 4 shifts of the grid (all the grid points form a lattice $Z^2/2$), how often is an individual node active?

5x5 network



Quasi-Regular Networks

Node usage:



We can show that:

- more than 50% of the nodes are used exactly once
- more than 25% of the nodes are not used.

Conclusions for Network Architecture

- In random networks, nearest-neighbor communication is very inefficient, and coverage can be a problem.

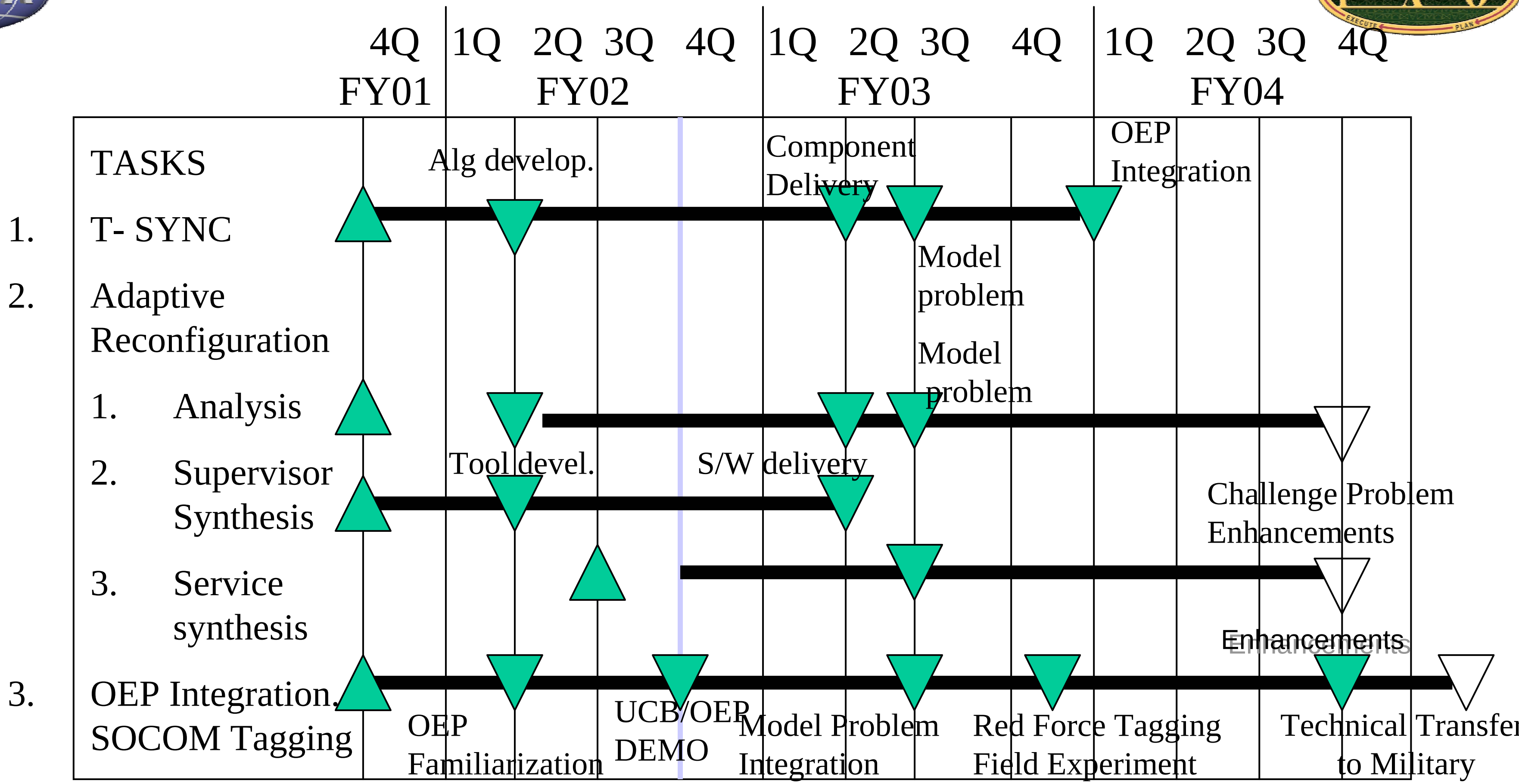
- To achieve a more regular topology, there are two equivalent possibilities:
  - Deploy nodes manually, approximately in a lattice
  - Use a subset of nodes from a random distribution

- So, accuracy in the deployment can be traded off by additional hardware.

Outline

- Red Force Tagging
- Large Networks of Embedded Sensors
- Program Issues
  - Milestones, future plans, issues

Project Schedule and Milestones



- On schedule.
- SOCOM Challenge Problem Enhancements.
- Robust TagMote and Middleware services. Wireless Networks-power issues

Technical Transition/Transfer

- Transition Activities
  - Military, SOCOM
- Opportunities
  - Universities

Program Issues

- Communication issues in Large scale networks:

It is known that ad hoc networks with random source-destination pairs scale badly.
It is crucial to identify communication schemes, architectures, and traffic patterns that overcome these limitations.

Real-time Configuration of Networked Embedded Systems

P.J. Antsaklis, Martin Haenggi, M.D. Lemmon
Dept. of Electrical Engineering
University of Notre Dame

DARPA/IXO NEST PI MEETING
December 2004, Tampa FL

Real-time Reconfiguration of Networked Embedded Systems

University of Notre Dame

Contract No. F30602-01-2-0526

AO No: L546/00

Period: 5/2001 – 5/2004. No Cost Extension to 12/2004

Panos J. Antsaklis (PI)

Department of Electrical Engineering

PI Phone No: 574-631-5792, Fax 574-631-4393, antsaklis.1@nd.edu

DARPA Program Manager: Vijay Raghavan

Agent name & organization: William Koenig, AFRL/Rome

Subcontractors and Collaborators

Senior Personnel
Panos J. Antsaklis, Michael D. Lemmon, Martin Haenggi
Electrical Engineering, University of Notre Dame

Post-doc
Marian Iordache

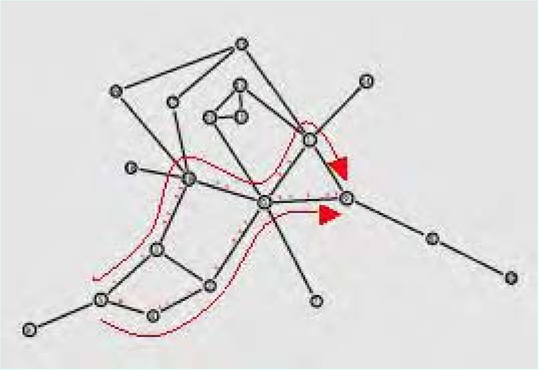
Graduate Students
Luis Montestruque, Lei Fang,
Xiaowen Liu, Nicholas Kottenstette

Univ. of Notre Dame

Real-time Reconfiguration of Networked Embedded Systems

Problem and Challenge

Real-time reconfiguration of middleware services in networks of embedded systems



Real-time+ Distributed Algorithms + Feedback Loops+Probabilistic Analysis+Formal Verification

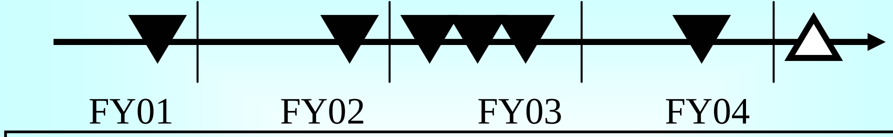
New Ideas

- Model-based clock synchronization and security for heterogeneous networks of embedded systems
- Adaptive reconfiguration for robust scalable middleware services in response to abrupt network changes
- Real-time network management through probabilistic analysis and formal verification.
- Exploiting feedback in distributed algorithms
- Interference Analysis of Large Networks

Impact

- Middleware for ad hoc networks of embedded systems.
- Real-time reconfiguration of streaming connections in ad hoc wireless networks.
- Red Force Tagging
- Large Network Architectures:
  - Quasi-regular networks
  - Interference characterization

Schedule



3QFY01: Project Start

3QFY02: Berkeley OEP Demo

1QFY03: Demo of Model Problem (robot formation control)

2QFY03: Delivery of TinyOS Service Components

3QFY03: Integration with Group I (Red Force Tagging)

3QFY04: Red Force Tagging & Networking Enhancements

1QFY05: Enhancements & Technical Transfer to Military

Problem Description and Objectives

Adaptive reconfiguration in fine grained networks of embedded systems that interact tightly with the physical world so to maintain Quality of Service (QoS.)

Project Tasks:

- Clock-synchronization:
  Develop clock-sync algorithms for ad hoc networks.

- Adaptive Reconfiguration
  Fast reconfiguration of NEST services
  in the presence of abrupt network changes

- Red Force Tagging for SOCOM

Project Status

PROGRESS SINCE LAST MTG.

- Red Force Tagging
  - Complete TagMote HW and SW enhancements

- Theoretical Study:
  - Analytical characterization of the local throughput in large interference-limited sensor networks
  - Throughput maximization and comparison of different channel access schemes

- Preparation of Final Report

Goals and Success Criteria

- **Red Force Tagging**
  - Final 2-Processor Version
  - Final Versions of Algorithms and Software

- **Large Networks of Embedded Sensors:**

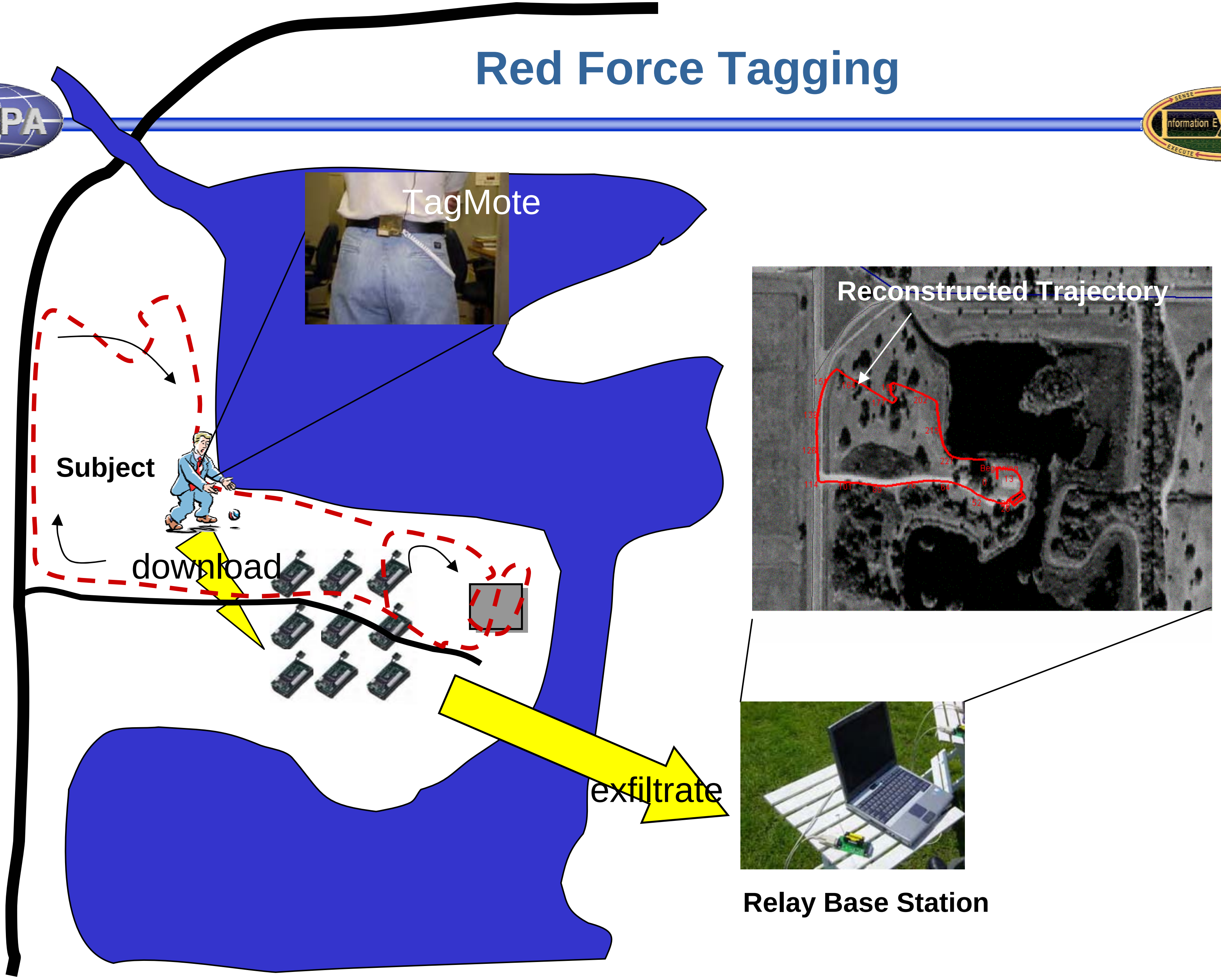
  Maximize throughput and lifetime.

Outline

- Red Force Tagging
  - Brief Project Description
  - TagMote HW and SW -- Final

- Theoretical Study: Large Networks of Embedded Sensors

Red Force Tagging



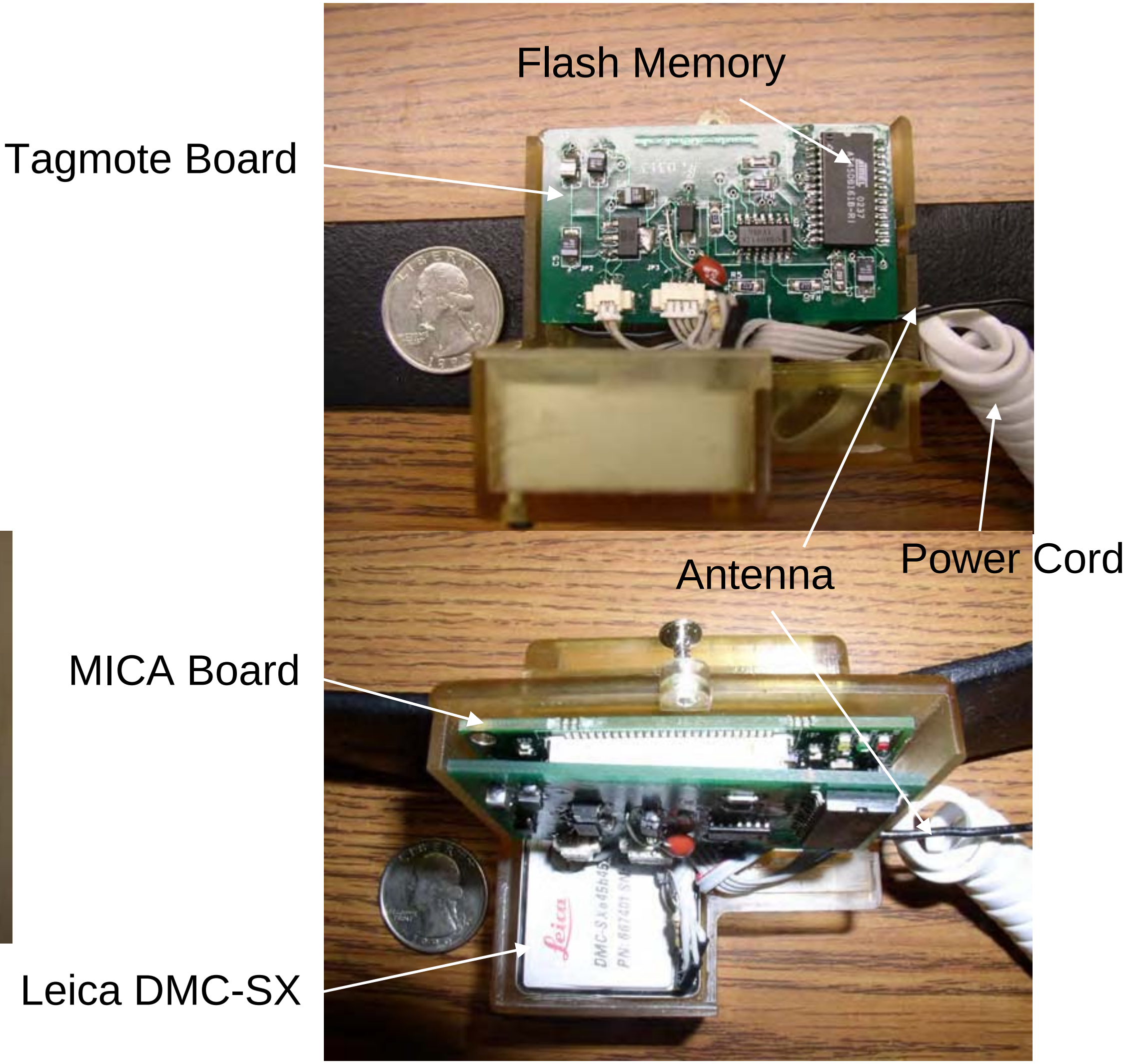
Relay Base Station

Benefits

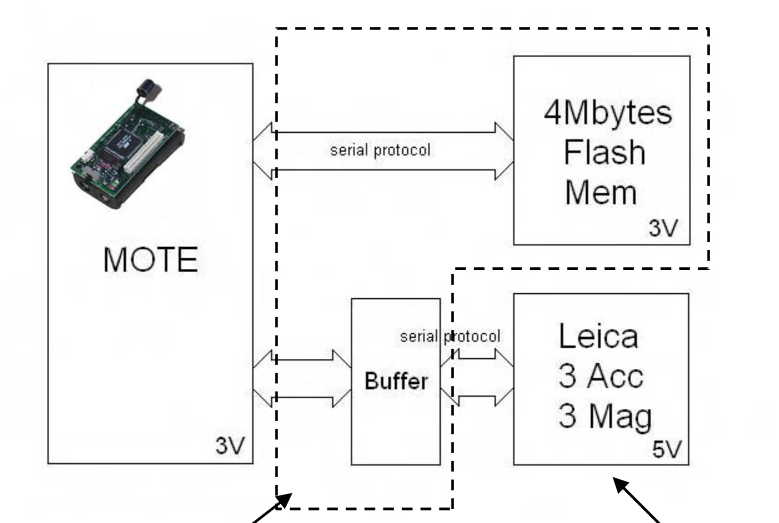
- Discovering Red Force in GPS-denied areas
- Does not rely on pre-deployed wired network
- Can be used by Blue Force
- Multi-sensor data collecting platform
  - Extensible to other sensor types : biochem, radiation

TagMote: Design 1.2 (single processor)

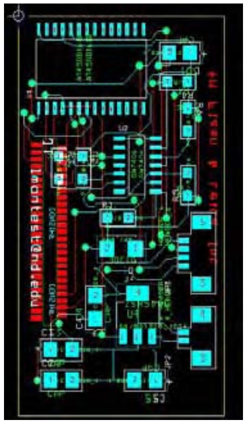
- Compact: 65 ×58×24 mm
- Lightweight:100 grams



TagMote Hardware (single processor)



Leica Module



TagMote Board

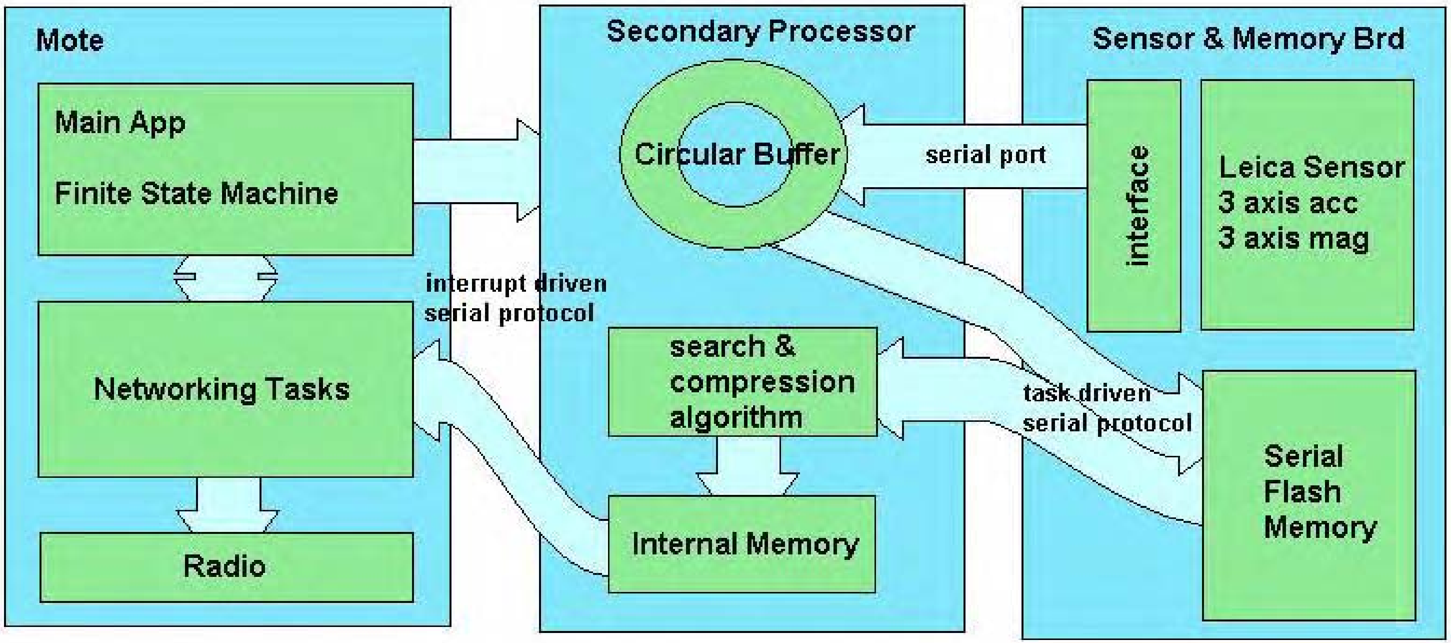
TagMote HW Redesign

- Increased speed of download
- Increased flexibility
  - Easier to add additional sensors
  - Easier to introduce more efficient algorithms, add new processing functions
- Keep same processor type for easy interface

TagMote Final HW

- Original TagMote Hardware:
  - Mica 2
  - Sensor and Flash Memory Board
- Final TagMote Hardware incorporates additional processor:
  - 8 MHz ATMEGA 128L
  - Standard Interface to Mica2 programmer
- Additional Processor (Slave) communicates with Mica2 (Master) using custom synchronous serial interface.
- Slave controls Sensor and Flash Memory card.
- Master in charge of Networking tasks

New HW Design-Adding Microprocessor



TagMote New HW

Master Tasks:

- Coordinate overall behavior.
- Find Network (ping).
- Identify node to download to.
- Forwarding of data to network (dump).

Slave Tasks:

Configure Leica Sensor.

Acquire data from Sensor.

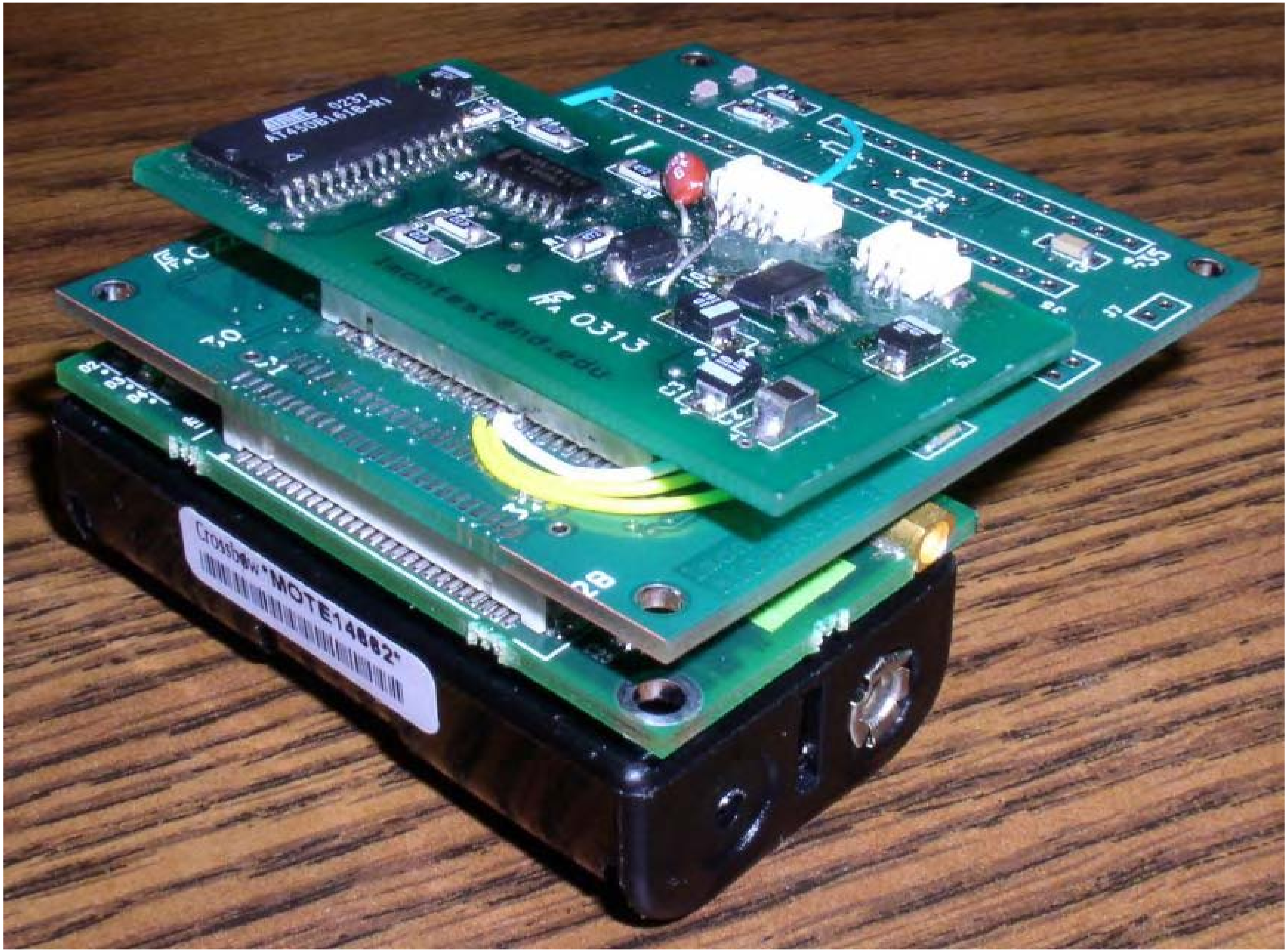
Process data (min max search).

Compress data.

Storing data in Flash.

Forwarding data from Flash to Master when needed.

TagMote HW: Design 2.0 (two processors)



Data Compression

- In the previous versions, we used 16 bytes of data:
  2 bytes for each of Gx, Gy, Gz, Ax, Ay, and Az

- In the new differential version we use 24 bytes:

Bytes 0 ... 3 contain: Time of data set 1 (28 bits),

data type: min or max (1 bit), packet type (compressed or not, 1 bit), differential time of data set 2 (2 bits).

The 32 bit word of bytes 0 ... 3 is unique, also used as msg ID

Bytes 4...23 contain: the differential measurements.

Enhancements

- Data compression increases speed of download and exfiltation and enhances length of walk stored in flash up to 16 hours

- For a 3 min walk, 45 sec download from TagMote to Network and 25 sec exfiltation to RelayMote

- Enhanced Download Service
  - ARQ-N scheme
- Enhanced Backbone service provides a highly connected routing structure to exfiltrate TagMote data
  - Keeps track of “retries” as estimate of link reliability
  - Keeps track of multiple “next hops” in a mote’s neighborhood

TagMote Software Tasks

- **TagMote software implements:**
- Compensation for adverse sensor orientation, calibration, filtering;
- Collection of data from sensors and storage on a non-volatile memory;
- Retrieval of data from memory, processing and transmission;
- Detection of NEST and data download;
- Pre-processing of raw sensor data before data download.

- **NEST network services provide**
- Synchronization: establishes global time reference;
- Backbone generation: establishes min-hop route to RelayMote;
- Streaming service: streams large blocks of data from NestMote to RelayMote;
- Download service: streams large blocks of data from TagMote to NestMotes.

- **Trajectory Post-Processing implements**
- Java programs on PC connected to RelayMote;
- Reassembly of data packet;
- Compensation for missing data;
- Reconstruction of complete trajectory;
- Display of trajectory over area map.

TagMote Software

- In addition to some of the NEST middleware components, described in the next subsection, the tag mote includes:
  - components that monitor the hardware
  - a communication component (The communication component allows a distributed implementation of the software on two processors)

The tag mote hardware components are:

□• **HWTAGMOTE**: This component is in charge of orchestrating all the hardware components and providing an interface for the top-level application. Real-time processing of sensor data is also done in this component.

- **FLASH**: This component manages the Serial Flash chip.
- **MODEB**: This component generates the appropriate command sequences for the Leica sensor that sets the Leica in continuous measurements of three accelerometers and three magnetometers. It also decodes the data sent by the Leica sensor.

□• **UARTPACKET**: It receives the data to be sent back and forth to the Leica Sensor. It receives data from the Leica in a byte-by-byte fashion, strips the frame from its overhead and forwards the payload data to the MODEB component. It also receives the frames to be sent to the Leica and transmits them byte by byte to the sensor through the UART component.

□• **UART**: It’s the component that manages the UART hardware, it initializes the hardware and receives and transmits the bytes from and to the UART.

NEST Middleware

- The NEST middleware consists of a number of services for network communication. Each of the tag mote, relay mote, and network motes uses some of the middleware services..

- The **broker service** builds a spanning tree in which the relay is the root and the network motes appear as nodes or leaves.
- The **dump service** is the name of the tag mote service that sends data to a network mote and also the name of the service of the network motes that stores locally the data transmitted by the tag mote.
- The **neighborhood service** maintains a list of the (active) neighbors a mote has.
- The **ping service** is the service used by the tag mote when it seeks to connect to the network.
- The **publication service** transmits the data stored in the network motes to the relay mote.
- The **route service** provides means of detecting the state of the network.
- The **synchronization service** is used by the network and relay motes to create a global clock.
- The **telemetry service** is useful for debugging purposes..

Base Station (Relay) Software

- The base-station software consists of java tools and the executable *undif*, implementing a C program.

- A java tool, called NestConsole is responsible of saving the data packets received from the relay mote in a file. That file is then read by undif, which decompresses the data. Finally, the decompressed data is processed by a java tool for processing and displaying the data.

Outline

- Red Force Tagging
- Throughput in Large Sensor Networks
- Program Issues

Throughput in Large Sensor Networks

Motivation: Event-based sensor networks are interference-limited when an event occurs. Sensor networks with period traffic are interference-limited since transmission rates must be kept low for maximum lifetime.

Objective: Mathematical analysis of the local throughput of large sensor networks. Comparison of different types of networks and MAC schemes.

The following network properties are modeled as deterministic or stochastic.

- Node positions: Deterministic of homogeneous Poisson point process.
- Channel gains: Path loss d^α with and without block Rayleigh fading. In the case of Rayleigh fading, the large-scale path loss is multiplied by an exponential RV with mean 1.
- Channel access: Deterministic (TDMA) or iid Bernoulli with parameter p (slotted ALOHA).

So, up to three sources of uncertainty are considered.

Throughput in Large Sensor Networks

- A packet is received if $\gamma > \Theta$, where γ is the (instantaneous) signal-to-interference ratio. Θ is a transceiver-dependent threshold.
- The success probability of a transmission is $p_s = \mathbb{P}[\gamma > \Theta]$.
  In Rayleigh fading, $p_s = \mathbb{E}[e^{-\Theta I}] = \mathcal{L}_I(\Theta)$!
- The local throughput g is the success probability times the probability that the transmitter transmits and the intended receiver listens.
- For slotted ALOHA with transmit probability p: $g = p(1-p)p_s$.
  For TDMA (every m-th node transmits): $g = p_s/m$.
- We want to take advantage of spatial reuse. The *spatial efficiency* of a network (and MAC scheme) is captured in a parameter σ.

For slotted ALOHA, the success probability can be expressed as:

$$\boxed{p_s = e^{-p/\sigma} \,,}$$

where σ is the spatial efficiency.

Throughput in Large Sensor Networks

Examples for the spatial efficiency (in slotted ALOHA):

Infinite random networks:

Without fading, $\alpha = 4$: $\sigma = 2/(3\pi\sqrt{\Theta})$.

With fading:

$$\sigma = \frac{\alpha}{2\pi\Theta^{2/\alpha}\Gamma(2/\alpha)\Gamma(1-2/\alpha)}$$

For $\alpha = 2$, $\sigma = 0$ (no spatial reuse).
For $\alpha = 4$, $\sigma = 2/(\pi^2\sqrt{\Theta})$.

Deterministic networks with fading. n potential interferers at distances r_i.

$$\sigma = \frac{1}{\sum_{i=1}^{n} \frac{1}{1+r_i^\alpha/\Theta}} \, .$$

In this case, σ is the “parallel connection” of $1 + r_i^\alpha/\Theta$.
The larger the distances r_i and α, and the smaller Θ, the better is the spatial efficiency.

Throughput in Large Sensor Networks

For regular line networks, the spatial efficiency is $\sigma = 1/(\pi\sqrt{\Theta} - 1)$ for $\alpha = 2$, and $\sigma = 1/(\pi\Theta^{1/4}/\sqrt{2} - 1)$ for $\alpha = 4$.

In general, for random networks: $\sigma^{-1} \propto \Theta^{2/\alpha}$.
For deterministic line networks with $r_1 = 1$: $\sigma^{-1} = c\Theta^{1/\alpha} - 1$.

Throughput maximization for slotted ALOHA:

$$g = p(1-p)e^{-p/\sigma}$$

is maximized for

$$p_{\text{opt}} = \sigma + \frac{1}{2}\left(1 - \sqrt{1 + 4\sigma^2}\right) .$$

Note that $0 < p_{\text{opt}} < 1/2$ and $0 < g_{\max} < 1/4$. The upper bound is achieved for $\sigma \to \infty$, the lower bound for $\sigma = 0$.

Throughput in Large Sensor Networks

Throughput maximization in TDMA:

- Regular line network, every m-node transmits.
- Distances are increased by a factor of m. Since only r_i^α/Θ matters, this means that rather than increasing the distances we can *decrease* Θ by m^α.
- Set $p = 1$ in the expressions for slotted ALOHA.

For slotted ALOHA and $\alpha = 2$:

$$p_s = \frac{\sinh\left(\pi\sqrt{\Theta(1-p)}\right)}{\sqrt{1-p}\sinh\left(\pi\sqrt{\Theta}\right)}\,.$$

With $p = 1$ and $\Theta \to \Theta m^{-\alpha}$:

$$p_s = \frac{y}{\sinh y}\,, \quad \text{where } y := \frac{\pi\sqrt{\Theta}}{m}\,.$$

The throughput $g = p_s/m$ can be expressed as

$$g = \frac{1}{m^2}\cdot\frac{\pi\sqrt{\Theta}}{\sinh\left(\frac{\pi\sqrt{\Theta}}{m}\right)}\,.$$

Throughput in Large Sensor Networks

With $\sinh x \approx e^x/2$, we find

$$m_{\text{opt}} = \lceil \frac{1}{2}\pi\sqrt{\Theta} \rceil \, .$$

For the achievable throughput, the very sharp approximation

$$g_{\max} \approx \frac{8}{\pi e^2 \sqrt{\Theta}} \approx \frac{0.34}{\sqrt{\Theta}}$$

can be derived.

For $\alpha = 4$:

$$m_{\text{opt}} = \lceil \frac{\sqrt{2}}{3}\pi\Theta^{1/4} \rceil \, ,$$

and the maximum achievable throughput is

$$g_{\max} = \frac{108}{\sqrt{2}\pi e^3 \Theta^{1/4}} \approx \frac{0.6}{\Theta^{1/4}} \, .$$

Throughput in Large Sensor Networks

Concluding Remarks:

- Comparison of slotted ALOHA and TDMA: Slotted ALOHA achieves 40%-50% of the TDMA throughput. $p_{\text{opt}} \approx 1/m_{\text{opt}}$, so the energy consumption is comparable but p_s is larger for TDMA.

- The throughput of all practical MAC schemes lies in between these two values.

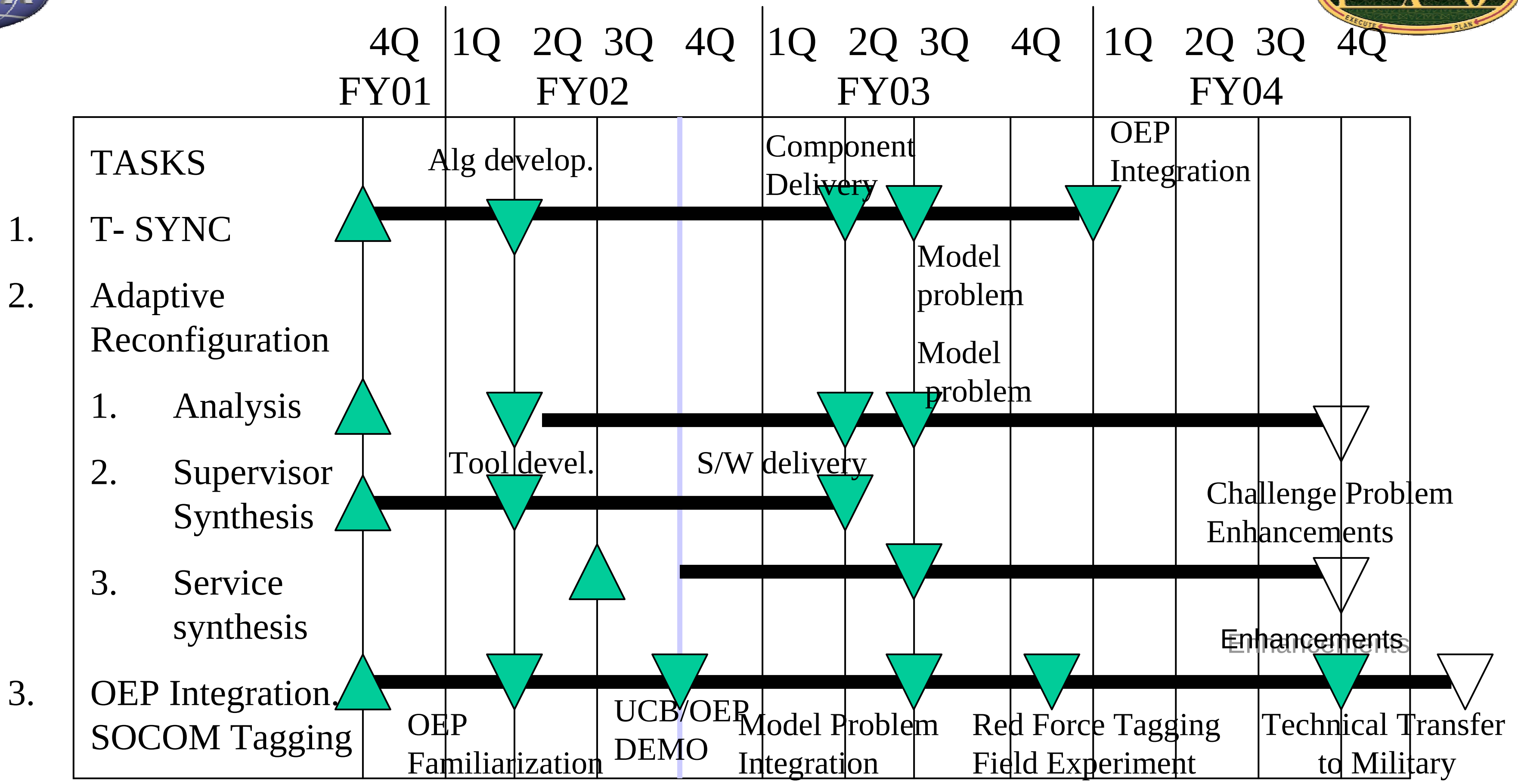
- The (local) throughput is only one performance metric. A detailed queueing analysis reveals that the end-to-end delay, in particular its variance, is affected much more strongly by random access. A *bounded-delay* packet dropping strategy helps keep the variance acceptable.

- Since $p_s = \mathbb{P}[\gamma > \Theta]$ is a complementary cumulative distribution, we know all the statistics of γ and the interference. This is important, since the mean interference for SIR is *not* sufficient to characterize the network. In fact, both are often infinite (not just for $\alpha = 2$ or fading networks) due to the heavy tails of their distributions.

Outline

- Red Force Tagging
- Large Networks of Embedded Sensors
- Program Issues
  - Milestones, issues

Project Schedule and Milestones



- On schedule.
- SOCOM Challenge Problem Enhancements.
- Robust TagMote and Middleware services. Theoretical studies of large-scale wireless networks.

Technical Transition/Transfer

- Transition Activities
  - Military, SOCOM

Program Issues

- Communication issues in Large scale networks:

  Interference is critical. It is essential that the interference is thoroughly characterized, and that meaningful performance metrics (e.g., spatial efficiency) are used to compare MAC schemes.